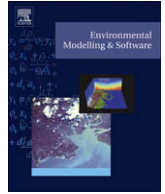




Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Formal verification of wastewater treatment processes using events detected from continuous signals by means of artificial neural networks. Case study: SBR plant

Luca Luccarini^{a,*}, Gianni Luigi Bragadin^b, Gabriele Colombini^a, Maurizio Mancini^b, Paola Mello^c, Marco Montali^c, Davide Sottara^c

^aENEA – ACS PROT IDR – Italian National Agency for New Technologies, Energy and the Environment, Environment Department, Water Resource Management Section, Via Martiri di Monte Sole 4, 40129 Bologna, Italy

^bDISTART, Faculty of Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

^cDEIS, Faculty of Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

ARTICLE INFO

Article history:

Received 16 October 2008

Received in revised form

10 April 2009

Accepted 18 May 2009

Available online xxx

Keywords:

Business process management

Event detection

Artificial neural networks

Intelligent systems

Rule-based management system

SBR

ABSTRACT

This paper proposes a modular architecture for the analysis and the validation of wastewater treatment processes. An algorithm using neural networks is used to extract the relevant qualitative patterns, such as “apexes”, “knees” and “steps”, from the signals acquired in the reaction tanks. These patterns, which show changes in the signals trend, are mapped to events in the process and logged using an appropriate XML format. The logs, in turn, are considered traces of the execution of a manufacturing process and validated using tools commonly applied for the Verification of Business Processes. The system has been applied to the data collected from a Sequencing Batch Reactor (SBR) for municipal wastewater treatment, equipped with probes for the on-line acquisition of signals such as pH, oxidation–reduction potential (ORP) and dissolved oxygen (DO). A SBR has turned out to be a suitable case study since the commonly acknowledged criteria for monitoring the biological processes (nitrification and denitrification) can be expressed in the form of qualitative constraints, which are easily translated into formal rules. The process logs, hence, are matched against these rules, which act as filters and quality classifiers.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Thanks to the recent developments in automation technology, nowadays instruments, control elements can be used to implement strategies and policies to handle most processes and problems in wastewater treatment plants (WWTPs), sometimes even increasing the capacity of biological nutrient removal by 10–30%.

The advanced knowledge acquired on the relationships between the operational parameters in a treatment system and the biochemical reactions (and thus its performance), gives new possibilities to control them, in order to improve the quality of the effluent while keeping the operational costs as low as possible. Exploiting these relationships the improvement due to Instrumentation Control Automation (ICA) may reach 20–50% of the total system investments within the next 10–20 years (Olsson, 2006). Energy efficiency is important as wastewater treatment industry is a competitive sector and energy costs typically consume from 15 to

30% of a treatment plant's operation and maintenance budget. An energy-management program may act on demand-site opportunities, including process modifications, aeration control and power-demand shift (Water Environment Federation, 2006).

The reasons why ICA technology is not universally adopted within WWT community vary between different countries, most of them are related to poor legislation, lack of acceptance within the WWT industries, lack of collaboration between stakeholders and organisations, economy and unreliable measuring devices; but probably the most fundamental barrier for a widespread acceptance of new control strategies is that existing WWTPs are not designed for real-time control. Within the EU it is a generalization to state that most WWTPs (>10000 p.e.) are equipped with SCADA (Supervisory Control and Data Acquisition) systems and PLC (Programmable Logic Controllers), and, even when present, they may still be used mainly for data acquisition rather than for the control of the operating conditions (Jeppsson et al., 2002). The degree of control (number and type of controlled variables, complexity of the strategies, etc...) that is cost efficient is dependent on size of plant. In Danish WWTPs, for example, in small plants (<10000 p.e) typical operating costs are in the range of 0.3–0.7 €/m³ of treated water. For plants over 250.000 p.e. the corresponding costs are 0.1–0.3 €/m³ of treated water. The

* Corresponding author. Tel.: +39 51 6098587; fax: +39 51 6098309.

E-mail addresses: luca.luccarini@enea.it (L. Luccarini), gianniluigi.bragadin@unibo.it (G.L. Bragadin), gabriele.colombini@enea.it (G. Colombini), maurizio.mancini@unibo.it (M. Mancini), paola.mello@unibo.it (P. Mello), marco.montali@unibo.it (M. Montali), davide.sottara2@unibo.it (D. Sottara).

difference in operation costs is dependent on the relative higher load variations at small plants than at larger plants, but also on the fact the instrumentation installed on smaller WWTPs is less effective, if present at all (Olsson et al., 2005).

Most treatment plants have several computers installed that collect data coming from sensors dislocated on the plant, but they are seldom actually used in real-time plant management. During abnormal conditions in WWTPs, success depends on the plant staff's ability to quickly identify the problem, diagnose it and start appropriate recovery actions. An intelligent control system could act as a "virtual expert operator", monitoring the processes continuously, and it could try to optimize the yield and detect faults at an early stage, possibly even correcting them. Then, the collected time series, properly validated and classified, could be used to build a knowledge base describing the various operating conditions of a given plant: this knowledge could be used to further improve the overall performance of the plant. Such a system, in fact, should be able to discern known situations, extracting features and patterns from the signals, and apply domain knowledge to assess the plant's health status and advise or choose itself the most appropriate control actions, effectively acting as a Decision-Support System (DSS) with Fault Detection and Isolation (FDI) capabilities (Olsson and Newell, 1999; Water Environment Federation, 2006).

Basically, intelligent control systems use AI-inspired tools to better control complex processes: these methods can be coarsely divided in two classes, symbolic knowledge-based architectures and soft-computing techniques. Knowledge-based approaches use heuristic knowledge and human experience to reason about the problem and include rule-based systems (Belanche et al., 1992; Giarratano and Riley, 1998) and case-based reasoning systems (Kolodner, 1993; Aamodt and Plaza, 1994). Soft-computing approaches include several techniques to solve nonlinear problems and include neural networks (Haykin, 1999) and fuzzy logic (Dubois and Prade, 1980; Klir and Folger, 1988). All these intelligent control techniques have been tested in wastewater treatment plants with interesting results.

Today, a new intelligent control systems, called intelligent decision-support system and based on combination of different artificial-intelligence tools, is emerging. Poch et al. (2004) propose a methodology for designing and building environmental intelligent decision-support systems, while many other published works show that this kind of systems can be implemented in both full-scale (Rodríguez-Roda et al., 2002) and pilot-scale plants (Baeza et al., 1999).

Sequencing Batch Reactors (SBR) are natural candidate targets for an experimental application of this kind of technology, since such plants allow can operate in highly customizable, different conditions without the need of structural modifications. The main advantage of SBRs is their great flexibility, which makes it possible to use these batch systems in a large variety of situations. SBRs are no more than batch reactors, but because of the nature of many industrial effluents, they are quite appropriate for wastewater treatment. Automation of SBR operations has made their implementation much easier and has definitely contributed to the development of SBR technology. A SBR has a unique cyclic batch operation, usually with five well-defined phases: fill, react, settle, draw, and idle. Most of the advantages of SBR processes can be attributed to their single-tank design and their ability to adjust the durations of the different phases, which endows these processes with a flexibility that allows them to meet many different treatment objectives. It is clear today that, for small and medium-sized wastewater treatment plants, SBRs should be seriously considered, not only for economical reasons, but also for efficiency in the removal of organic matter and nutrients, provided that a previous

and proper design is used (Artan and Orhon, 2005; Irvine et al., 1997; Mace and Mata-Alvarez, 2002).

The SBR cycle for the treatment of municipal wastewater usually requires a simple sequence of two biological processes. The first is the denitrification process, that occurs in anoxic conditions, and the second is nitrification process in aerobic conditions (Henze et al., 2000). The automatic detection of the end of these processes allows to optimize the duration of the operational cycles, which usually would be set on a worst-case basis, in order to reduce costs. In fact, anoxic conditions should be maintained only as long as the denitrification reaction is taking place, and aerobic ones only for the time necessary to let nitrification happen. If the process is not monitored, the only possibility is to fix the duration of the phases to a value which is usually a compromise between reaction yield and number of treatment cycles per day. The deadlines are usually chosen considering the time it would take to process a heavy load batch, which is usually higher than the average case. This conservative choice guarantees that, assuming the plant works properly, the concentration of pollutants in the effluent will be low, but at the cost of time and energy, especially to keep the tank in aerobic conditions. Moreover, if the process is not monitored, nothing can ensure its effectiveness, i.e. that the effluent is conforming to the required standards.

In the years, several criteria have been proposed to detect the end of the reactions, based on neural networks (Cho et al., 2001; Luccarini et al., 2002; Akin and Ugurlu, 2005; Aguado et al., 2009), on fuzzy clustering (Marsili-Libelli, 2006), on fuzzy logic (Marsili-Libelli et al., 2008) and on a combination of neural networks and fuzzy logic (Cohen et al., 2003); Yoo et al. (2004) used multiway principal components analysis as a monitoring tool to detect abnormal conditions. Another hybrid system for real-time monitoring and control appears in (Sottara et al., 2009), where a possible way of combining different modules is also shown. All these criteria are conveniently based on analysis of chemo-physical indicators such as pH, oxidation-reduction potential (ORP) and dissolved oxygen concentration (DO) since the sensors for their measure are cheap and easily available.

The main aim of this work is to propose yet another set of tools for the intelligent management of treatment processes, with the SBR being a special case that also served as test case.

An analysis algorithm, based on a peculiar use of neural networks, has been developed and used to identify the position, the extension and other features of local events in the signals. These events, such as "apexes", "knees" and "steps", denote changes in the trends of the signals and can be used to describe and analyze the evolution of a signal from both a qualitative and a quantitative point of view. The events, in fact, model the signal at a higher level than the samples themselves and can be used as input for other analysis modules in a more general architecture. To cite one, such events can be mapped on "Episodes" as proposed by (Sánchez-Marré et al., 2005) and be integrated in a case-based reasoning schema.

In this work, however, yet another approach is proposed. The low-level signal events are used to define and detect some higher level process events, such as the completion of a reaction or a change in operating conditions. Then, an analogy is formed between these events and the production steps in a manufacturing process. This allows to define a mapping between the treatment processes and the different stages of a Business Process (BP) (Weske, 2007). The topic of Business Process Management (BPM) has received much attention from the scientific community in the last years, which led to the development of several tools useful for the management, the optimization and the validation of such processes. In particular, this work deals with process validation: the events detected by the signal analysis module are used to trace

the evolution of the treatment process. This trace, then, can be checked for compliance against a set of business rules describing the expectations and the requirements of an “optimal” operative condition. The rules themselves define the concept of optimality: in the specific case of an SBR cycle, a set of rules has been generated to verify the actual completion of the required biochemical reactions.

2. System architecture

The proposed architecture is composed by several cascaded modules. At the lowest level lies the plant itself, where the processes take place. The probes immersed in the tank collect the samples with a 1-min frequency, which are then pre-processed before being stored in a database. Pre-processing includes filtering out unrealistic samples, typically due to probe malfunctioning, denoising and tagging the time series for later access.

The next module processes the acquired signals individually, extracting the local features required to detect the signal-level events such as the cited maxima. The only and obvious requirement is that the signals show regular and smooth trends, which changes can be described in qualitative terms. In particular, the following set of events has been considered and identified (Spagni et al., 2001):

- **Step(T)**: a signal, constant in some interval $[T - \delta_l, T]$ drops or rises from a level L_{old} to a level L_{new} and remains constant thereafter for some time $[T, T + \delta_r]$. A step may be rising or falling, depending on whether $L_{new} > L_{old}$.
- **Apex(T)**: the signal reaches a local maximum or minimum, showing a falling (resp. rising) trend in $[T - \delta_l, T]$ and a rising (resp. falling) trend in $[T, T + \delta_r]$.
- **Knee(T)**: the signal shows a falling (resp. rising) trend which tends to a constant value in $[T - \delta_l, T]$ ($dS(t)/dt > 0$ and $dS(t)/dt \rightarrow 0$ as $t \rightarrow T$). At time T , a new falling (resp. rising) trend appears as $|dS(t)/dt|$ steps to a larger value.

All the events have a positive (rising) or negative (falling) version. After they have been detected, criteria involving the signal-level events are applied to detect higher, process-level events. The assumption is that a combination of some specific signal-level events is correlated to a relevant change in the status of the process and thus can be used to identify it.

Eventually, exploiting the analogy between process states and manufacturing stages, a treatment cycle is considered a special case of business process, so it can be traced, classified and possibly validated according to the number and type of identified process-level events using the standard Business Process Management techniques.

Fig. 1 shows the flow of information between the different modules. In this work, the process-level events will only be used for validation, while future works will address the topic of using the available information to control the state of the process, or at least to support the decisions of the plant operator.

2.1. Plant requirements

The proposed architecture does not depend strictly on the type of plant, as long as the process can be observed, at least partially, sampling specific chemical and physical parameters, such as the cited pH, ORP and DO, but also temperature, conductivity, SST (suspended solids concentration) and so on. The sampled time series are required to have relatively slow trends and changes in these trends are to be somewhat related to changes in the internal state of the treatment process. Hence, oscillatory signals (or any signal which would be likely analyzed in the frequency domain) are not directly suitable for the proposed type of analysis.

However, the class of candidate plants is still large and includes continuous flow activated sludge plants and Membrane Biological Reactors (MBR), other than the already cited SBRs.

2.2. Signal-level events

The detection of patterns in signals is performed using Neural Networks, albeit in a novel way.

Neural networks are commonly used for recognition tasks and a feed-forward network could be used to detect the presence of an event in a time windows. However, the events of interest are actually classes of events. Consider, for example, a local maximum: different shapes exist, depending on features such as the steepness, the symmetry and the amplitude (i.e. $[T - \delta_l, T + \delta_r]$). To ensure that a network trained to recognize maxima would have sufficient generalization capabilities, a large training set composed of different examples would be needed and even then there would be no guarantee on the network performance but the comparison between its output and the expected results on a known test set. Given these difficulties, a different approach has been used: a simple network is trained and optimized to approximate the desired signal; afterwards, it is considered as a white-box model and its parameters are analyzed to extract the desired information.

Notice, however, that the modularity of the general architecture allows any event detection module to be plugged in.

2.2.1. Pre-processing

In order to analyze a given signal $S(t)$, a temporal window has to be filled with the collected sample. When validating a SBR process, a logical choice is taking $W \approx 300$ min, one for each sample at the standard rate. It corresponds to the full length of a cycle minus the time the reactor remains empty. However, when performing real-time detection, a growing window or a sliding time window of fixed length should be used. Before being processed, the acquired signals are checked to ensure that the data lie within sensible ranges (in the specific case, 5–9 for pH, -500 mV– 500 mV for redox potential and 0–12 for DO), to exclude probe malfunctioning and consequently notify the plant operator. In case of malfunctioning, the data could still be plausible according to such filters, but the detection procedure will likely fail to report any significant event. This in turn will affect the output of the validation rules, which try to address this problem as discussed in the relevant section.

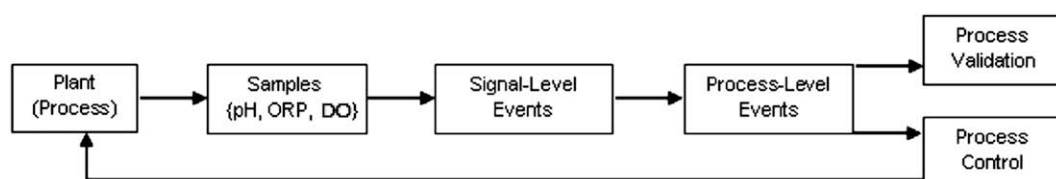


Fig. 1. System architecture.

Under normal circumstances, instead, the data in the time window are de-noised, using a regularization algorithm (Chartrand, 2007) which returns a clean time derivative of the original signal and, by integration, a cleaner and smoother version of the signal itself.

2.2.2. Neural network interpolation

The de-noised samples are normalized and processed by a feed-forward neural network with sigmoidal and linear hidden and output activation functions, respectively, which tries to learn the essential information contained in the data. This type of network is a well-known function approximator (Li et al., 2002): in this case, the relation is a map from the Time domain T to the Signal range, the set of real values \mathcal{R} which, due to normalization, can in practice be restricted to the interval $[-3,3]$. The data in the time window forms a (small) training set of pairs $\{t, S(t)\}$ (where S denotes a generic signal at time t), which is used to train a 1-N-1 network.

The size N of the hidden layer is a critical parameter, since a value too high would lead to an overfitting of the original signal, while a network with a value too low would not be able to approximate the data. To choose an optimal value, a procedure based on the Haar wavelet decomposition has been adopted. A signal $S(t)$ is iteratively subsampled into its approximation and detail descriptors (Donald and Walden, 2000), so that at the i -th iteration the sample at time t can be computed recursively:

$$S_i(t) = \frac{S_{i-1}(2t) + S_{i-1}(2t + 1)}{\sqrt{2}} \quad (1)$$

The approximation coefficients at different levels form a tree, which leaves are the original signal. However, any cut of the tree can be considered an approximation of $S(n)$, the more detailed the deeper the border nodes are chosen in the tree. Each cut C has an associated energy, according to equation:

$$E_C = \frac{\sum_i S_i(t)^2}{\sum_t S(t)^2} \quad S_i \in C \quad (2)$$

Starting from the trivial cut including only the root node, the tree is explored greedily, replacing, at each iteration, a node with its two children, choosing the node that causes the greatest increase in the energy E_C .

The procedure stops when a sufficient fraction α of the total energy (usually $\alpha = .99$) has been obtained, yielding a cut which is deeper, and thus more detailed, in zones where the original signal shows more variations. The cut is used to define a linear approximation of S and the size of the hidden layer N is chosen to be equal to the number of tracts.

In fact, in the chosen architecture, each neuron N_j is defined by three parameters A_j, w_j and b_j : A_j denotes each of the N hidden-output weights, w_j the N input-hidden weights and b_j the N biases. The output $y(t)$ of the network, thus, is a linear combination of the activations of the N hidden neurons:

$$y(t) = \sum_{j=1}^N A_j \cdot \tanh(w_j \cdot t + b_j) + y_0 \quad (3)$$

Taken individually, each sigmoid is centred on the instant $C_j = -b_j/w_j$, has an extension which is proportional to $E_j = 1/w_j$ and an amplitude A_j due to the hidden-output weight, as shown in Fig. 2. Each sigmoid actually contributes non-constant information in the interval $I_j = [C_j - kE_j, C_j + kE_j]$ ($k \approx 2.5 \div 3$), so it is feasible to approximate a sigmoid with a linear model which points are

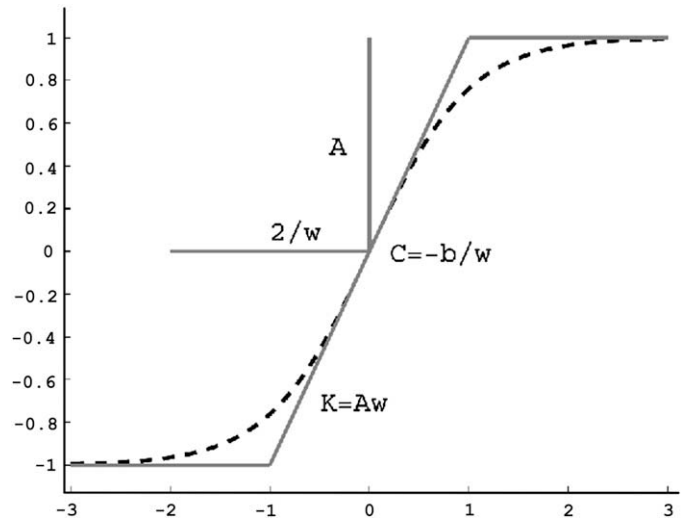


Fig. 2. Neuron features.

biunivocally defined by the same three parameters: in particular, the MSE between a sigmoid and its approximation is minimized for $k \approx 1.25$.

Exploiting and inverting this relation, the linear model used to define the structure of the network can also be used to initialize the network parameters before using the gradient back-propagation training algorithm, and is an alternative to the standard random initialization. This heuristic usually ($\approx 70\%$ over 1165 cases) allows the training to be completed in less epochs and even then in no more than a few ones ($\leq 10 \div 20$).

Given the initial layout, the intervals in which each neuron is not constant do not overlap, so there is a biunivocal relation between each neuron and a tract of the original signal: hence, the trend of that each tract is determined only by the parameters $\{A_j, w_j, b_j\}$ of some neuron N_j .

This is a convenient property to analyze the signal, so the training procedure should not cause the intervals I_j to overlap considerably. Hence, the network is trained using the quadratic error function and the back-propagation algorithm: at each iteration, however, the error function is regularized (Haykin, 1999) by an additional term:

$$E(n) = (S(n) - y(n))^2 + \alpha \sum_{j=1}^N (w_j - w_j^0)^2 \quad (4)$$

As shown in Bishop (2006), such a regularization term constrains each weight w_j to keep values near its initial w_j^0 , preventing overlap and forcing each neuron to keep modelling only local trends.

The final MSE is usually negligible ($MSE/|\max(S) - \min(S)| < 0.01$), but if it exceeds that threshold the procedure is considered failed since no good approximation of the signal could be found. This is useful since the signals that can't be approximated correctly are usually non-smooth and symptom of some anomaly in the process.

2.2.3. Network organization

The number of neurons deployed and thus the level of detail in approximation depends strongly on the energy threshold α . The chosen value ($\alpha = .99$) has been determined experimentally to capture most trends in signals, but even using the adaptive procedure it leads to a number of neurons which is too high. Hence,

Table 1
Event property definitions.

	Time	Duration	Type
Step	C_j	$[(C_j - kE_j), (C_j + kE_j)]$	$\text{sign}(A_j)$
Apex	$A_j w_j C_j + A_{j+1} w_{j+1} C_{j+1} / A_j w_j + A_{j+1} w_{j+1}$	$[C - \min((C - C_j), (C_{j+1} - C)), C + \min((C - C_j), (C_{j+1} - C))]$	$\text{sign}(A_j)$ if $\text{sign}(A_j \cdot A_{j+1}) = -1$
Knee	$C_{j+1} - 1/w_{j+1}$	$[C - \min((C - C_j), (C_{j+1} - C)), C + \min((C - C_j), (C_{j+1} - C))]$	$\text{sign}(A_j)$ if $\text{sign}(A_j \cdot A_{j+1}) = +1$

a final pruning procedure has been developed to merge redundant neurons. In many cases, in fact, a uniform trend is split into smaller ones of similar slope which could still be modelled by a single neuron.

The algorithm uses the linear approximation again, since the join of two segments can be easily defined using only the four extreme points while it's not so trivial how to do it given two sigmoidal functions.

```

remove neurons with  $A_j < \varepsilon \approx 0$ 
given a pair of neurons  $N_1 = N_j$  and  $N_2 = N_{j+1}$ :
  compute gain intervals  $I_{i,1,2} = [L_i, R_i] = [C_i - kE_i, C_i + kE_i]$ 
  set  $X \leftarrow \text{sort}([L_1, L_2, R_1, R_2])$ 
  set  $X \leftarrow X - X[1]$ 
  set  $Y[0] \leftarrow 0, Y[3] \leftarrow A_1 + A_2$ 
  if  $(R_1 < L_2)$  set  $Y[1] = Y[2] = A_1$ 
  else
    set  $Y[1] \leftarrow K_1 X[1]$ 
    set  $Y[2] \leftarrow Y[1] + (K_1 + K_2)(X[2] - X[1])$ 

```

linearly interpolate the points (X, Y) , getting correlation coefficient ρ_{ij}^2 .

The segment obtained at step 6, clamped at levels 0 and $A_1 + A_2$, gives the candidate approximation for a neuron N^* which could replace N_j and N_{j+1} , with the constraint that the correlation coefficient must be higher than a chosen threshold ρ_0 . If that is not the case, the two neurons capture different trends and have to be kept. In particular, a compatibility vector $M(j)$ is computed for each pair of adjacent neurons, with the values being equal to ρ_{ij}^2 : at every iteration, the pair with the highest value is merged and the vector is updated, until there is no suitable pair left.

At the end of the procedure, the number of components is reduced (in the tests, the number of neurons usually went from ~ 20 to ~ 10 , depending on the particular signal), but the locality property is preserved, since the interval of the union is roughly the union of the original intervals. Thus, each neuron in the new set still models a different tract of the signal.

2.2.4. Event definition

Since the events of interest are actually changes in signal trends, and, if the locality property is satisfied, different trends are modelled by different neurons, it is feasible to detect the presence of an event from the analysis of adjacent pair of neurons alone.

- **Step:** A step can be modelled by a single sigmoid, so the detection of an isolated neuron, i.e. for which $\max(|(C_j - kE_j) - (C_{j-1} + kE_{j-1})|, |(C_{j+1} - kE_{j+1}) - (C_j + kE_j)|) > \Delta$, allows to detect this class of events.
- **Apex:** Two adjacent neurons with discordant gains, i.e. for which $|(C_{j+1} - kE_{j+1}) - (C_j + kE_j)| < \delta$ and $\text{sign}(A_j) \neq \text{sign}(A_{j+1})$, are used to identify such events.
- **Knee:** The presence of an event of this type may be denoted by two adjacent neurons with different, concordant gains, i.e. for which $|(C_{j+1} - kE_{j+1}) - (C_j + kE_j)| < \delta$ and $\text{sign}(A_j) = \text{sign}(A_{j+1})$.

By definition, events are not instantaneous, so all events are also defined by the following properties (Table 1):

- **Time:** most representative moment t_{event} , which is typically an estimate of the instant of maximum change in trend.
- **Duration:** time interval $[t_{\text{start}}, t_{\text{end}}]$ in which the effects of the event manifest themselves. Notice that events are not necessarily symmetric, so in general $t_{\text{event}} \neq (t_{\text{start}} + t_{\text{end}})/2$.
- **Type** $\in \{+1, -1\}$: allows a distinction between maxima and minima, and between rising and falling steps and knees.

The events are detected using the local approximation and not the original signal, so an additional check is performed: considering the interval $[t_{\text{start}}, t_{\text{end}}]$ alone, the Pearson's correlation coefficient between the original signal and the (two) sigmoid(s) involved is computed. This coefficient is preferred to other indicators such as the MSE since it is not necessary to approximate the signal exactly, but to model its trends alone. This coefficient measures the confidence in the detection of the event: if it falls below a threshold equal to 0.85, the event is discarded as it is likely spurious.

2.3. Validation using process-level events

A process-level event is a higher level event, such as the start or the completion of a reaction in the plant. Typically, the effects of a process-level event manifest as lower level events in one or more of the monitored signals: the detection of certain event patterns in the signals has been used extensively to identify the process events, especially in SBRs where this correlation is almost universally acknowledged (Zhao et al., 1999; Kim et al., 2004).

The conditions which support the recognition of an event can be defined using reactive business rules expressed in the language CLIMB (Chesani et al., 2008), which has been proposed for the task of validating (business) processes. The rules have an "IF Body THEN Head" structure: the conditions in the Body define a context which, when verified, leads to the expectation that the conditions in the Head should be satisfied as well. Hence, rule Bodies can be used to discriminate the type of plant and process in order to apply the appropriate validation criteria. The Heads, instead, list the requirements for a process to be considered valid in the desired sense, according to the listed expected properties.

To apply the validation procedure, a process is transformed into a sequence of (signal-level) events which become "audit trails" entries in a process log structured according to the MXML meta-model (Van Dongen and van der Aalst, 2005). MXML is an XML-based language for representing business process execution traces, generated by process-aware information systems during execution. The model defines some main properties which are adequate to describe the signal events according to the mapping:

```

<Source>, the plant where the treatment is performed
<ProcessInstance>, modelling each treatment period or cycle
<AuditTrailEntry>, each individual event:
<WorkflowModelElement/>, an identifier defining the
event: Step{+/-}, Apex{+/-}, Knee{+/-}.

```

<EventType/>: according to the MXML transactional model, events may denote the *start* or the *completion* of activities lasting in time, or the *performing* of a task.

<Originator/>, defining the source of the event: pH, ORP, DO.

<Timestamp/>, denoting the instant at which the event takes place.

</AuditTrailEntry>

</ProcessInstance>

</Source>

The signal analysis module produces a MXML log for each treatment process, with an entry for each detected signal event. A log is matched against one or more rules expressing relational conditions on the entries: a rule either accepts or rejects it, depending on whether the conditions are satisfied or not. A set of process instances, thus, is partitioned by a rule into two subsets, the “valid” ones and the “failed” ones.

In order to evaluate the rules concretely, a tool known as SCIFF checker¹ has been used (Alberti et al., 2008). This tool, which has been conveniently implemented as a plug-in of the ProM framework (Van der Aalst et al., 2007), supports the CLIMB rule language and can process the MXML logs, partitioning the instances and providing visual reports on the results. The resulting subsets can then be further processed by other rules, effectively applying them in cascade, to perform more complex validations.

In a typical scenario, one could use rules to define the conditions necessary to detect a relevant process-level event, such as the triggering or the completion of a certain reaction. In order for a process to be validated, however, it could be necessary not only to detect all the process events in a set, but also to impose expectation constraints on them. This could be easily accomplished if, whenever the same instance is validated by different rules, the outcome was registered in a meta-MXML log, so that additional meta-rules could be applied to this higher level information. The meta-event descriptors could use the standard tags with a semantic similar to the following one:

<AuditTrailEntry>, each individual event:

<WorkflowModelElement/>, event type: Success, Failure.

<EventType/>: *start* and *end* denote a period over which the originator rule is valid (or invalid). *Perform* denotes an instant evaluation.

<Originator/>, identifying the rule which generated the event.

<Timestamp/>, denoting the time at which the properties listed in the originator rule are considered true or false (which, in general, could be different from the time at which the rule is evaluated)

</AuditTrailEntry>

Unfortunately, even if conceptually feasible, this feature is not supported in the current implementation. The application to a given instance of different rules in cascade or in parallel is equivalent to composing the rules with standard Boolean and- and or- logic connectives, without the type and time constraints that can be expressed in the CLIMB language.

3. Case study: SBR pilot plant

3.1. Plant configuration

The data have been collected from a pilot-scale SBR, built by ENEA and located next to the municipal plant of Trebbo di Reno

(BO, Italy). The plant, which has a working volume of 500 L and overall hydraulic retention time (HRT) 20 h, is fed with real sewage drawn after grit removal, whose peak load characterization is presented in Table 2. Biomass is automatically wasted daily to maintain the sludge retention time (SRT) between 15 and 20 days. The plant is operated with 4 cycles a day, consisting in a sequence of feed, reaction (anoxic, then aerobic), sludge wasting, settling, draw and idle phases. It is equipped with a mechanical mixer, a variable-flow blower connected to a membrane diffuser, two peristaltic pumps for influent loading and effluent discharge (flow rate = 6 L/min) and a pump for sludge wastage (flow rate 1 = L/min). A multi-parameter converter continuously measures pH, ORP and DO signals during the whole process: the samples are acquired by a National Instruments data acquisition board and stored in a MySQL database for both immediate and later access. The actuators are controlled by a PLC interfaced to the acquisition board.

The input sewage characteristics variable both during the same day and in different days. These variations can be observed in the acquired signals, e.g. in Fig. 3, which trends do not repeat periodically with the same features.

In fact, depending on the characteristics of the influent, the signals show important trend changes that could be used for the diagnosis of the process in course. Some track studies were carried out, and one of them is reported as example in Fig. 4. The track confirms the theoretical expectations: the end of the denitrification process can be detected according to the maximum in the pH signal and the knee with negative slope in the ORP data. Likewise, the end of the nitrification process is marked by the point of minimum pH, the rising DO concentration and the knee with positive slope in the ORP curve.

3.2. Signal analysis

A database comprised of 1165 cycles has been assembled from data acquired during several periods from June 2004 to June 2007. Each data set is relative to a whole treatment cycle and is composed by 360 sampled values for each signal pH, ORP and DO. These data sets, stored in a MySQL database, have all been submitted to the system for an off-line classification and validation. The low-level event detection has been performed on each cycle using the proposed procedure, after excluding the last 60 samples relative to the settling, discharge and idle phases.

Applying the de-noising algorithm, the initialization heuristic, and the back-propagation with the customized objective function to the 300 samples, the training of each neural network could be performed in less than 100 epochs (most of the times in less than 20), yielding a trained network with an average of 20 neurons (10 in case of DO) in the hidden layer. After that, the pruning procedure usually reduced the neurons to about one third of the original number.

The example in Fig. 5 shows the different phases of the training of a network for the modelling of a pH time series: Fig. 5a shows a detail of the original signal and its de-noised version; Fig. 5b

Table 2
Peak load influent characterization.

Parameter	(mg/L)	St. Dev.	Method
COD _T	250–400	–	APHA 1992
COD _S	200–350	–	APHA 1992
pH	8.1	0.07	Potentiometric
TSS	135	0.02	APHA 1992
VSS	30	0.01	APHA 1992
N-NH ₄	60.4	6.8	APHA 1992
TKN	69.8	4.5	APHA 1992
BOD/COD	0.4–0.7	–	APHA 1992

¹ <http://www.lia.deis.unibo.it/research/climb/verification-Conformance.html>.

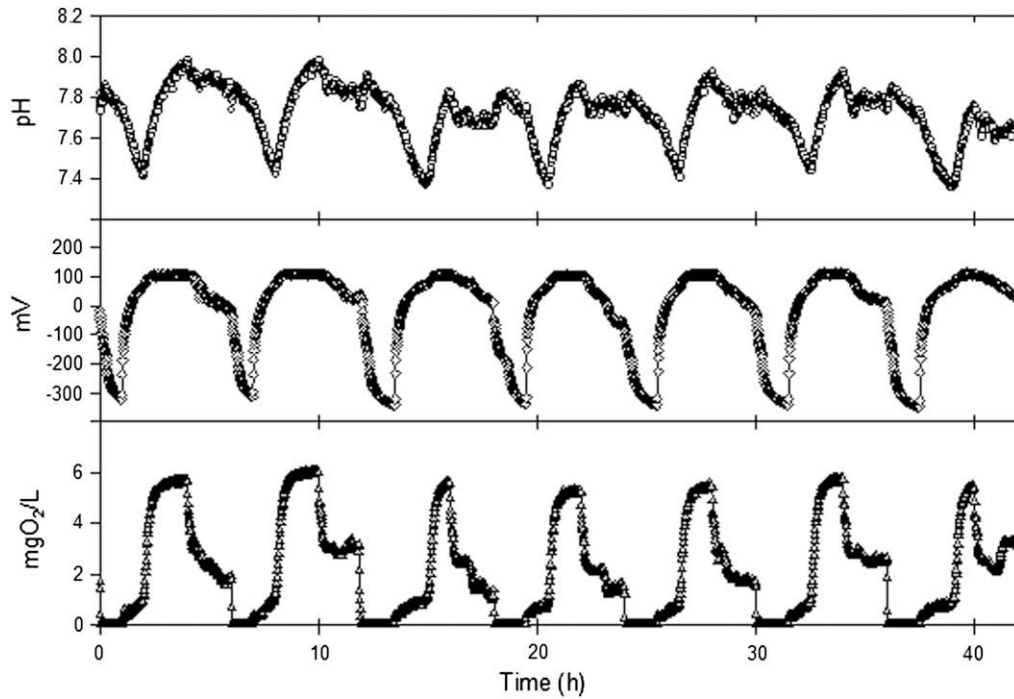


Fig. 3. pH, ORP and DO acquired during several consecutive cycles.

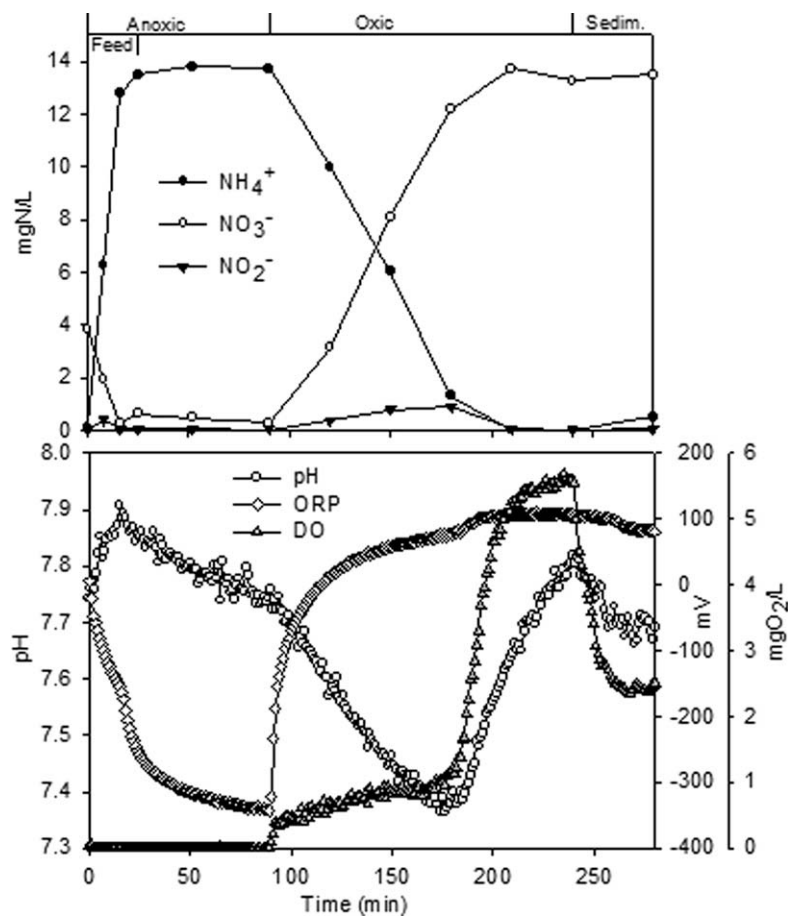


Fig. 4. Track study during a single treatment cycle in the SBR.

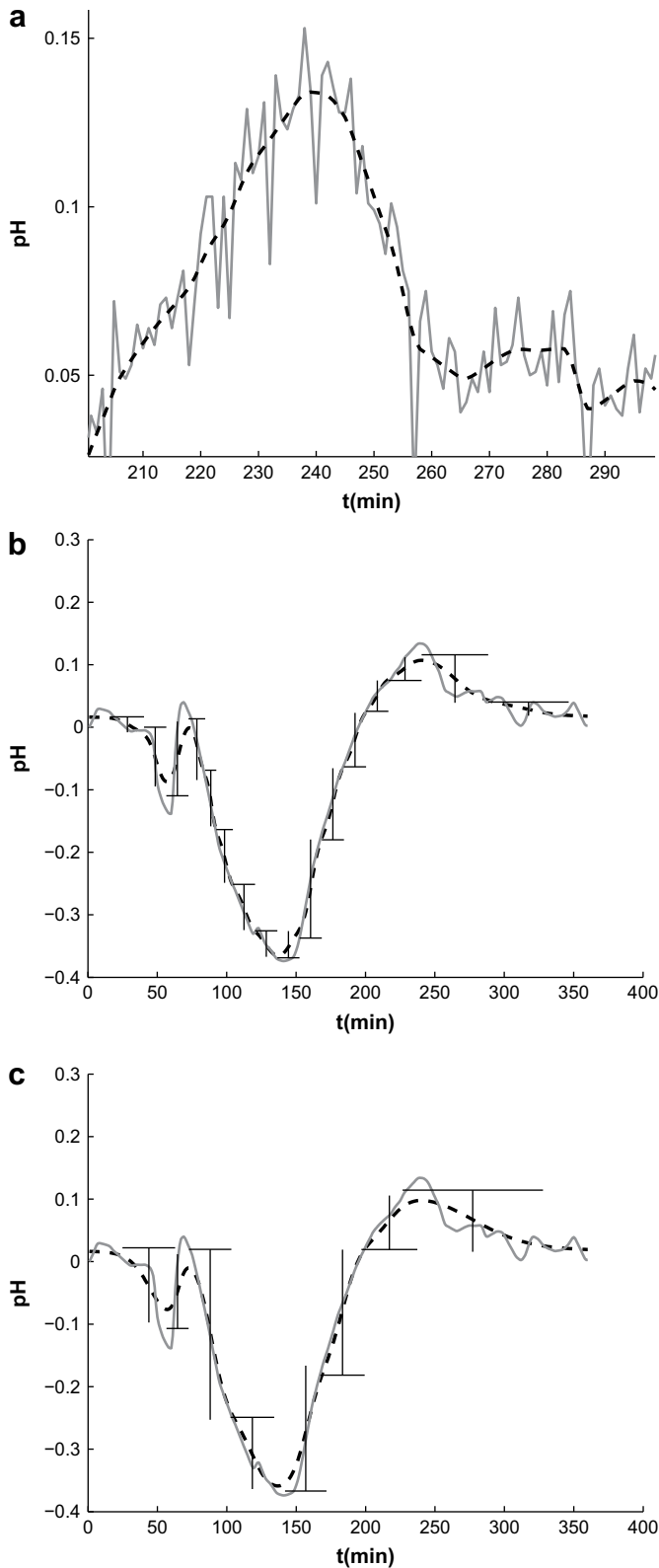


Fig. 5. a. De-noised signal (detail). b. Trained Neural Network ($N = 16$). c. Pruned Neural Network ($N = 8$).

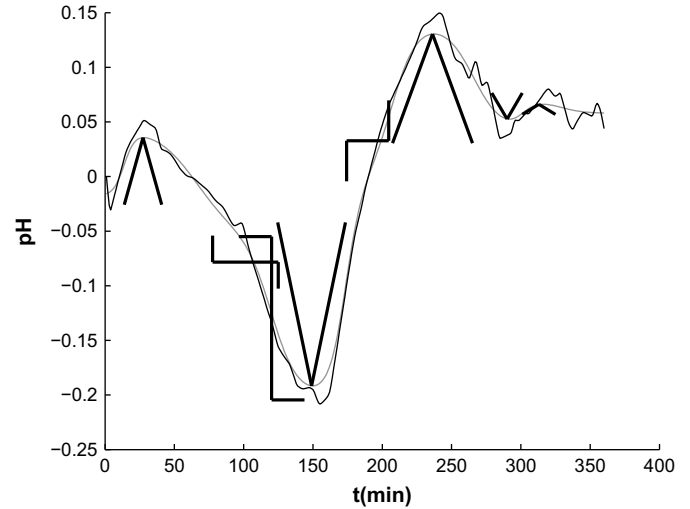


Fig. 6. Events detected in a pH time series.

shows the trained 16-neuron neural, while Fig. 5c reports the final, pruned network. The neurons are drawn using “T” shapes to show their extension ($C_j \pm 2E_j$) and amplitude A_j . Notice that the simplified network no longer approximates the signal exactly, but it still manages to reproduce its qualitative trends.

The parameters of the neurons are then used to detect the events: Fig. 6 shows the results for a pH time series. The events themselves are represented using V-shaped symbols for apices, and horizontal and vertical S-shaped symbols for knees and steps respectively. Size and orientation are proportional to extension and type.

3.3. Validation rules

3.3.1. Process event rules

The CLIMB language allows a direct mapping of the well-known criteria for the identification of the end of the reactions. As an example, consider the rule **D**:

D:	IF	activity A is started by Oa at time Ta
	having	A equal to Cycle
	and	Oa equal to SBR
	THEN	activity B should be performed by Ob at time Tb
	and	activity C should be performed by Oc at time Tc
	having	B equal to Apex+ and Ob equal to pH
	and	C equal to Knee- and Oc equal to ORP
	and	Tc after Tb - ΔT_{min} and Tc before Tb + ΔT_{min}
	and	Tb before Ta + D_{maxT}
	and	Tc before Ta + D_{maxT}

This rule states that a maximum in pH and a knee in ORP must be detected almost contemporarily within the deadline D_{max} allowed for the denitrification reaction. The ΔT_{min} window has been introduced to deal with noise due to the measures themselves and the interpolation error. In fact, the events are usually not perfectly aligned even in the original signals and the approximation process introduces an error which has to be taken into account.

With a similar formalism it is possible to define the criteria for the Nitrification and the switch between the anoxic and the aerobic phase, summarized here in a simplified form:

D:	End of Denitrification:	pH Apex ⁺ and ORP Knee ⁻
A:	Aerobic Switch:	pH Knee ⁻ and ORP Apex and DO Step ⁺
N:	End of Nitrification:	pH Apex ⁻ and ORP Knee ⁺ and DO Step ⁺

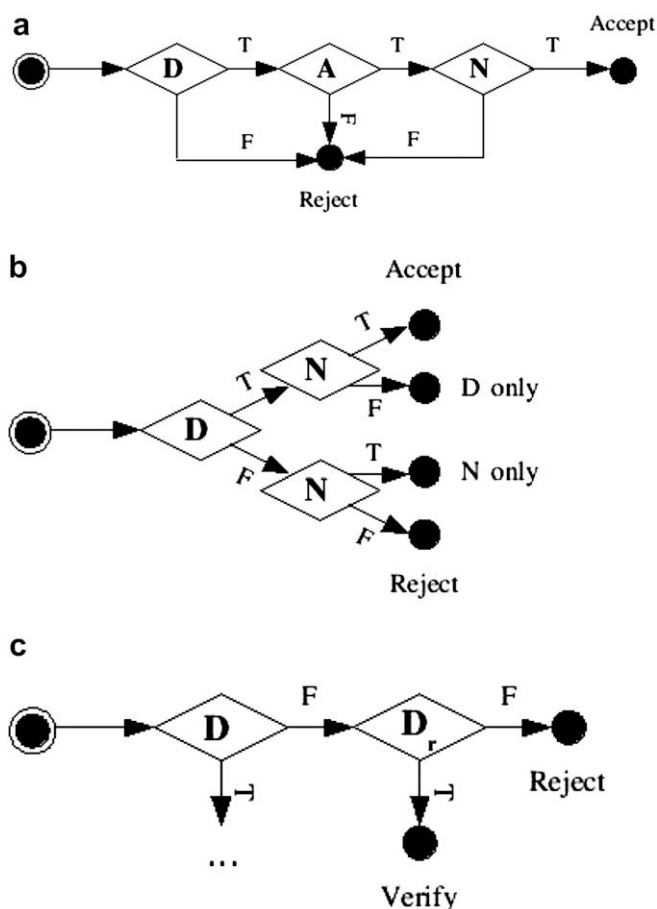


Fig. 7. a. Standard Validation. b. Classification. c. Coarser Validation.

Notice that the evaluation of condition **A** is not strictly necessary for optimization purposes, since the aerators are activated on command, but can nevertheless be used to verify the actual functioning of the air flow in the plant.

3.3.2. Cycle validation rules

Given the current capabilities of the SCIFF checker, rules can only be composed in flows. The basic policy, which can be applied both in and off-line, states that a cycle is to be validated if it is accepted by the rules **D**, **A**, **N**, as shown by the flowchart in Fig. 7a. This policy, however, does not allow distinctions between the reasons why a cycle can be rejected: to achieve this goal, different rule-flows have to be adopted.

If one is interested in dividing the cycles according to which reactions have been completed, it is possible to adopt the pattern in Fig. 7b which, using only rules **D** and **N**, assigns a given cycle to one of the four possible classes corresponding to both, either one or no reaction detected.

The main limitation of the given rules is that they validate only cycles with ideal profiles: in a SBR plant, cycles repeat regularly with clearly discernible features only in complete absence of perturbations. When dealing with a pilot-scale plant, different external factors such as variations in the influent load and external temperature or alterations in the bacterial population may affect the process. Such process noise adds to the measurement noise due to the probes, so the cycles are rarely completely compliant to the expected standards.

However, most cycles are still acceptable if one accepts to relax the constraints. From the observation of the available historical

data, several anomalies have been found in the signals. For example, the value of the ORP at the beginning of a cycle may be very low (below -250 mV), so that the value actually tends to rise before falling again: the signal, hence, shows a maximum instead of a knee when denitrification completes. Likewise, at the end of the nitrification reaction, pH usually reaches a minimum, but the other two signals rise by values which can be barely unnoticeable as well as quite large. This means that the change in ORP and DO can be modelled by a rising step, a rising knee or nothing at all, depending on the single case. Finally, when trying to detect the switch between the anoxic and the aerobic conditions, it turns out that the indicator event depends on the conditions at the end of the anoxic phase. pH is the most affected signal: under normal operating conditions it usually passes from a slowly descending trend to a faster one (modelled using a knee), but if denitrification is not complete, its pristine trend is constant. The discontinuity due to the oxygen flow usually causes it to fall (even if in some rare cases it caused it to rise), so the signal analysis is more likely to detect falling steps and maxima rather than negative knees.

These conditions are less desirable, but not sufficient to mark a cycle as completely invalid. The CLIMB language does not support graded or fuzzy constraints, so there is no way to express different quality degrees. However, a pattern such as the one in Fig. 7c can be used: a process instance is validated using the ideal rule, for example **D**. Any instance that fails to satisfy it is not discarded, but examined using one or more relaxed rules such as \sim **D**. Instances that satisfy the relax rule but not the strict one can be treated accordingly: the simplest policy is to accept them with a warning. A possible set of relaxed rules, in compact form, is the following:

\sim D:	pH Apex ⁺ and (ORP Knee ⁻ or ORP Apex ⁺)
\sim A:	(pH Knee ⁻ or pH Apex ^{+/-} or pH Step ⁻) and ORP Apex ⁻ and (DO Step ⁺ or [])
\sim N:	pH Apex ⁻ and (ORP Knee ⁺ or []) and (DO Step ⁺ or DO Knee ⁺)

3.4. Experimental results and discussion

In order to test the effectiveness of the proposed architecture, the signal processing algorithm has been implemented using Matlab[®]. Its output, MXML logs recording all the detected events, has been filtered using the SCIFF checker tool programmed with both the ideal rules **A**, **D** and **N** and their relaxed counterparts. Of the 1225 available cycles, 60 were discarded because they contained nothing but noise: the remaining 1165 have been validated by a human expert, who said whether the relevant process events (End of Denitrification, Aerobic Switch and End of Nitrification) could be detected from the observation of the signals. The expert's response has been compared with the classification returned by the checker.

For greater accuracy, the tests have been executed for different values of the threshold ΔT : 1, 2, 3, 4, 5, 7.5, 10, 12.5, 15, 20, 25, 30, 40, 50 and 60 min. The results are reported using ROC curves (Bradley, 1997): the opinion of the human expert is considered "correct", so the output of the classifier can be divided in the canonical true positive (TP), true negative (TN), false positive (FP) and false negative (FN) classes. Thus, the performance of the classifier can be measured using the true positive, as sensitivity (Sens), and true negative, as specificity (Spec), rates

$$\text{Sens} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$1 - \text{Spec} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad (6)$$

The position in the plane depends on the value of ΔT : the ROC curve shows the evolution as the threshold varies. A value of

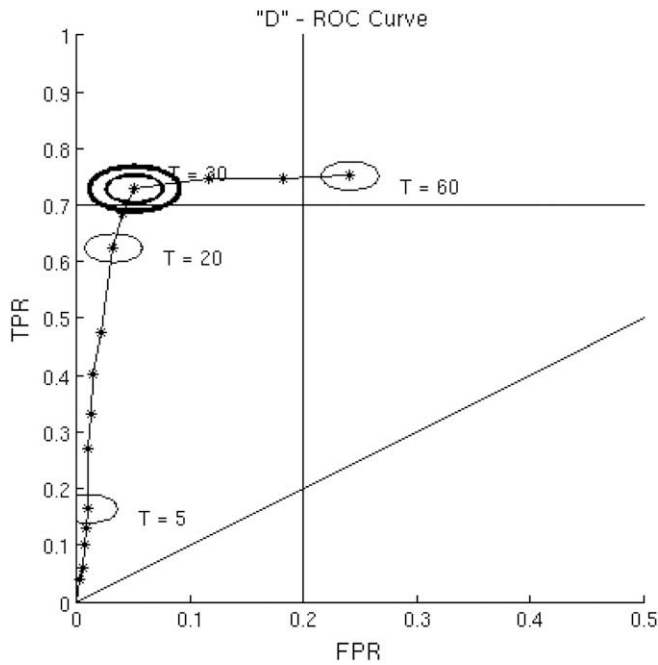


Fig. 8. ROC curve for rule "D".

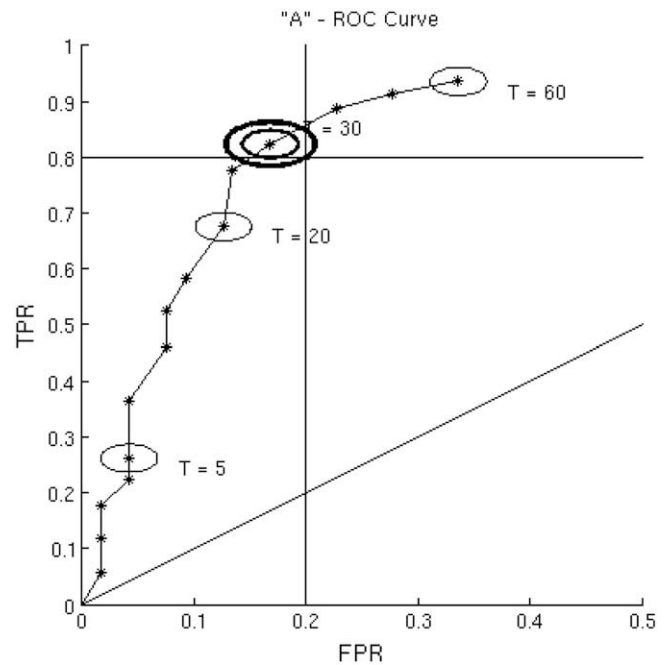


Fig. 9. ROC curve for rule "~A".

$\Delta T \div 20/30$ min brings the accuracy over 70% and is generally sufficient to deal with the noise and the timing error introduced by the estimation, which can be as high as 10 min. Moreover, Cohen's coefficient Kappa is calculated and reported to measure the degree of agreement between the software and the operator it tries to emulate.

Notice that, in general, the proposed classification method can't reach accuracy = 1, independently on the threshold of choice, since it possible for a required event not to be detected, so the validation rule will always fail.

3.4.1. End of denitrification

The analysis of the denitrification reaction shows two important results: the first is that the plant hasn't been able to complete the process in approximately one cycle out of two during its history. Moreover, the accuracy is lower than for other rules: the knee in the ORP isn't always evident, especially if compared with other events, so the analysis algorithm tends to overlook it. It is also difficult to define a relaxed rule since the DO level brings no information and the detection of the pH apex alone would lead to many false positives. In this case, the cost of a false positive is higher – pollutants are discharged – than the cost of a false negative – the reaction lasts longer than needed – so the cautious behaviour is preferred. The results are shown in Fig. 8 and in Table 3.

3.4.2. Anoxic phase switch

Being an externally induced state change, true negative failures are quite rare, but a true negative evaluation is a symptom of a serious malfunctioning which has to be dealt with. Due to the

Table 3 Confusion matrix for rule "D" ($\Delta T = 30$ m).

		Human Expert	
		T	F
SW	T	417	30
	F	155	563

Kappa = 0.681

various possible denitrification scenarios, however, the response of the system to the oxygen input is hardly determined univocally. Even using the relaxed rule(s) ~A instead of A, a positive response can be considered reliable while a negative one should lead to further investigations due to the high cost associated to an aeration system failure. The results are shown in Fig. 9 and in Table 4.

3.4.3. End of nitrification (relaxed)

In an ideal cycle, the completion of the nitrification reaction is denoted by a marked rising step in both ORP and DO. This, in practice, is seldom true since the former is often too small to be distinguished from noise and the latter may also be considered a rising knee rather than a step, especially if the reaction completes in a short time. The table, in fact, shows that the relaxed version of the rule is far more accurate. The results are shown in Fig. 10 and in Table 5.

3.4.4. Entire cycle validation

The validation of the whole cycle does not differ significantly when rule ~A is applied in addition to the mandatory D and ~N. The failures during the anoxic phase are the limiting factor, in addition to the limited capability of the system in detecting the faster trends of the signals during denitrification. The results are shown in Fig. 11 and in Table 6, for the rule "D ~ N", and in Fig. 12 and in Table 7, for the rule "D ~ A ~ N".

4. Conclusions and future developments

The different modules in the system (data pre-processing, event detection and cycle validation) form a flexible architecture for the

Table 4 Confusion matrix for rule "~A" ($\Delta T = 30$ m).

		Human Expert	
		T	F
SW	T	862	20
	F	184	99

Kappa = 0.407

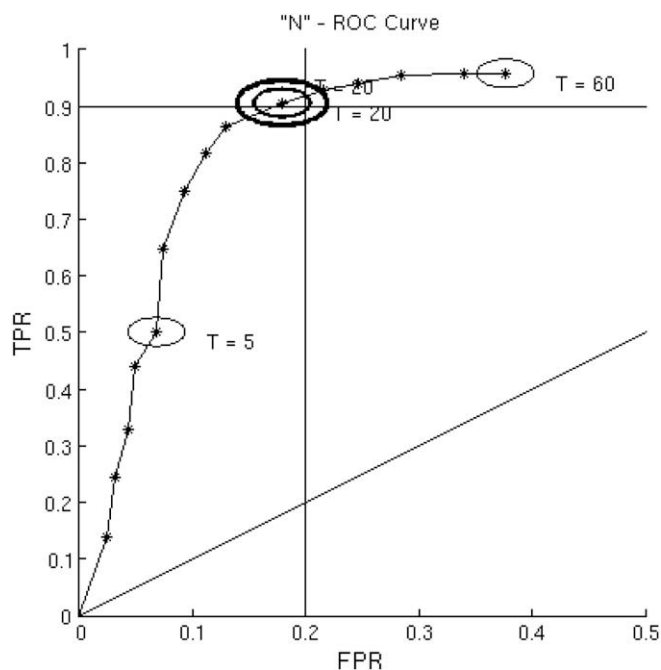


Fig. 10. ROC curve for rule "~N".

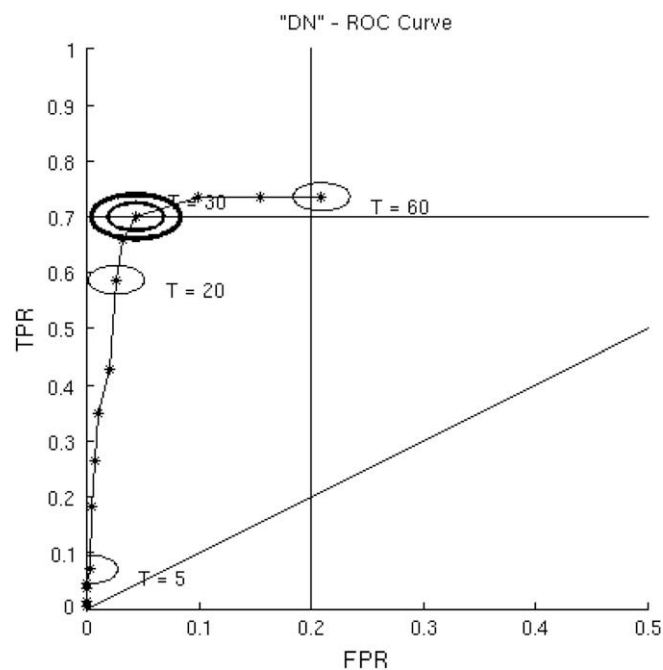


Fig. 11. ROC curve for rule "D ~ N".

logging and the analysis of treatment processes. In particular, it has been shown that a sequence of reactions in a wastewater treatment plant can be considered a verifiable "business" process, a first step for the application of concepts and techniques which are being developed with important results in other industrial contexts. Future studies will consider the applicability and usefulness of other business-related concepts than simple validation, such as scheduling, control and optimization, especially when applied to treatment plants which are more complex and operate at larger scales than a simple SBR pilot plant. In fact, in the near future some BPM techniques will be applied to the management of the municipal WWTP of Calderara di Reno (BO, Italy), in the context of a research project sponsored by HERA S.p.A.

The present activity on the SBR plant, instead, has shown many interesting points and limitations of the adopted approach. Using the proposed optimizations, the event detection algorithm could also be applied during a treatment process, assuming that the sampled data are stored in a buffer in real-time. The main benefit of this technique is that it does not limit itself to detect a feature, but tries to model it returning more information than a simple flag. At the moment, the position and the extension are provided along with a confidence degree, but more features are being defined, both qualitative and quantitative. The events, in fact, could be described in greater detail using fuzzy properties and parameters which take into greater account the variety of shapes that can be observed. Signals, in fact, show a high degree of variability, both due to measurements and especially to process

noise. For example, pH maxima, when present, have slopes ranging over a full order of magnitude, appearing like steep peaks or flat hills. Moreover, using a full set of features, a newly detected pattern could be matched against some typical archetypes mined from the historical data collected from the plant. Signal shapes such as maxima and minima do not depend on the nature of the signal, nor on the meaning they could have, but instances – or classes thereof – with particular characteristics could possibly be associated to some specific operating conditions in the context of a process or a plant.

The use of "imperfection" (fuzziness and probability) in the description could also make the detection module more robust and reliable. At the moment, in fact, the system either accepts completely a detected event or discards it, considering it spurious. Instead, a future implementation could try to analyze a signal again, using different threshold values, whenever the modelling error is too large to return meaningful information.

This additional information could also be beneficial for the rule-based validation system. Validating cycles or, more generally, processes is common practice since it allows the evaluation of intuitive criteria, which can be easily defined and understood from a human operator interacting with the process and, possibly, its control system. The SCIFF checker, however, doesn't take uncertainty into account, not even the confidence degree returned by the event detection module, so less reliable events are considered exactly like more reliable ones when trying to validate the cycles. Moreover, given the noise in the data and some approximation

Table 5

Confusion matrix for rule "~N" ($\Delta T = 20$ m).

		Human Expert	
		T	F
SW	T	436 (909)	8 (29)
	F	154 (94)	567 (133)

Kappa = 0.352 (0.623)

Table 6

Confusion matrix for rule "D ~ N" ($\Delta T = 30$ m).

		Human Expert	
		T	F
SW	T	377	27
	F	161	600

Kappa = 0.669

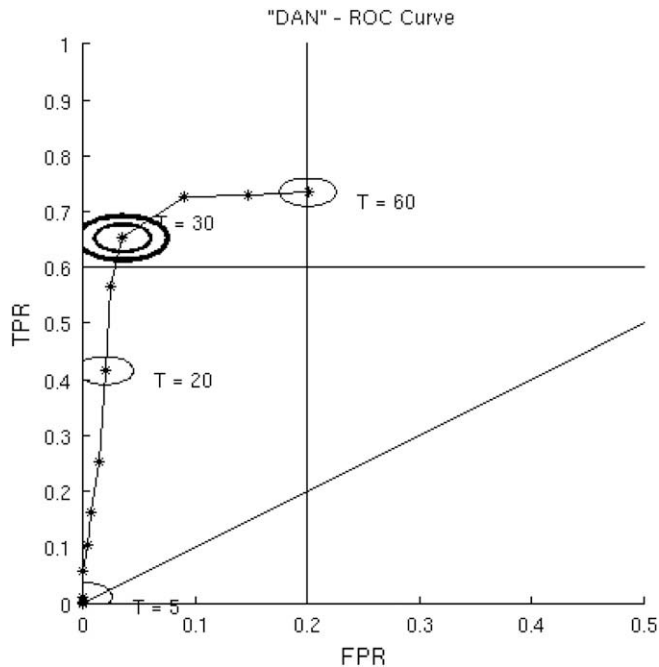


Fig. 12. ROC curve for rule "D ~ A ~ N".

error, events which are temporally aligned in the original signals aren't detected exactly at the same time. The threshold ΔT alone does not allow to distinguish between misalignments due to an approximation error from those which are effectively shifted in time.

Hence, it will be considered whether introduce these additional capabilities in the checker itself or to integrate it with other software tools.

The current proposed rules still manage to validate cycles at different levels of quality by including different types of constraints. The use of intermediate information such as signal-level events, in fact, allows to define rules with a sufficient degree of flexibility while providing a convenient abstraction level to avoid the complexity of expressing constraints on the values themselves. In fact, while the criteria expressed in the specific rules can also be used to control the SBR process, their implementation in the SCIFF checker makes them more suitable for validation purposes rather than real-time control. Several examples of other appropriate systems have been cited in the introduction, but the validation system could and will be integrated to verify the performance of the "intelligent", real-time control modules.

As a collateral application, the proposed system is useful in the development of other intelligent controllers: the partitioning capabilities of the checker could be useful to create training sets for supervised learning. In fact, rules could be used to define the properties of the desired training data and extract them from the database in a very efficient manner.

Table 7
Confusion matrix for rule "D ~ A ~ N" ($\Delta T = 30$ m).

		Human Expert	
		T	F
SW	T	350	22
	F	186	607
Kappa = 0.632			

Acknowledgements

The authors wish to thank the anonymous reviewers who have contributed to the improvement of this paper and HERA S.p.A. for the availability and hospitality inside its plants.

References

- Aamodt, A., Plaza, E., 1994. Case-fundamental issues, methodological variations and system approaches. *AI Communications* 7 (1), 39–59.
- Aguado, D., Ribes, J., Montoya, T., Ferrer, J., Seco, A., 2009. A methodology for sequencing batch reactor identification with artificial neural networks: a case study. *Computers and Chemical Engineering* 33 (2) 465–472.
- Akin, B.S., Ugurlu, A., 2005. Monitoring and control of biological nutrient removal in a sequencing batch reactor. *Process Biochemistry* 40 (8), 2873–2878.
- Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P., 2008. Verifiable agent interaction in abductive logic programming: the SCIFF framework. *ACM Transactions on Computational Logic*, 9(4), ACM New York, NY, USA.
- Artan, N., Orhon, D., 2005. Mechanisms and Design of Sequencing Batch Reactors for Nutrient Removal. IWA Scientific and Technical Report no. 19. IWA Publishers, London.
- Baeza, J., Gabriel, D., Lafuente, J., 1999. An expert supervisory system for a pilot WWTP. *Environmental Modelling and Software* 14 (5), 383–390.
- Belanche, B., Sánchez, M., Cortés, U., Serra, P., 1992. A Knowledge-based System for the Diagnosis of Waste-water Treatment Plants. *Lecture Notes in Computer Science, Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Springer-Verlag, Berlin.
- Bishop, M.C., 2006. *Pattern Recognition and Machine Learning*. Springer Science Publishing.
- Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30 (7), 1145–1159.
- Chartrand, R., 2007. Numerical differentiation of noisy, nonsmooth data. Available at: <http://math.lanl.gov/Research/Publications/Docs/chartrand-2007-numerical.pdf>
- Chesani, F., Mello, P., Montali, M., Riguzzi, F., Sebastianis, M., Storari, S., 2008. Compliance checking of execution traces to business rules: an approach based on logic programming. In: *Proceedings of the Sixth International Conference on Business Process Management*.
- Cho, B.C., Liaw, S.L., Chang, C.N., Yu, R.F., Yang, S.J., Chiou, B.R., 2001. Development of a real-time control strategy with artificial neural networks for automatic control of a continuous-flow sequencing batch reactor. *Water Science and Technology* 44 (1), 95–104.
- Cohen, A., Hegg, D., De Michele, M., Song, Q., Kasabov, N., 2003. An intelligent controller for automated operation of sequencing batch reactors. *Water Science and Technology* 47 (12), 57–63.
- Donald, B.P., Walden, Andrew T., 2000. *Wavelet Methods for Time Series Analysis*. Cambridge University Press.
- Dubois, D., Prade, H., 1980. *Fuzzy Sets and Systems. Theory and Applications*. Academic Press.
- Giarratano, J.C., Riley, G.C., 1998. *Expert Systems: Principles and Programming*, third ed. Course Technology.
- Haykin, S., 1999. *Neural Networks: a Comprehensive Foundation*, second ed. Prentice-Hall.
- Henze, M., Harremoës, P., la Cour Jansen, J., Arvin, E., 2000. *Wastewater Treatment—Biological and Chemical Processes*, third ed. Springer, Heidelberg.
- Irvine, R.L., Wilderer, P.A., Flemming, H.-C., 1997. Controlled unsteady state processes and technologies – an overview. *Water Science and Technology* 35 (1), 1–10.
- Jeppsson, U., Alex, J., Pons, M.N., Spaniers, H., Vanrolleghem, P.A., 2002. Status and future trends of ICA in wastewater treatment – a European perspective. *Water Science and Technology* 45 (4–5), 485–494.
- Kim, J.-H., Chen, M., Kishida, N., Sudo, R., 2004. Integrated real-time control strategy for nitrogen removal in swine wastewater treatment using sequencing batch reactors. *Water Research* 38 (14–15), 3340–3348.
- Klir, G.J., Folger, T.A., 1988. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall Int. Editions, New York.
- Kolodner, J., 1993. *Case-based Reasoning*. Morgan Kaufmann, San Francisco CA.
- Luccarini, L., Porrà, E., Spagni, A., Ratini, P., Grilli, S., Longhi, S., Bortone, G., 2002. Soft sensors for control of nitrogen and phosphorus removal from wastewaters by neural networks. *Water Science and Technology* 45 (4–5), 101–107.
- Li, H.X., Li, L.X., Wang, J.Y., 2002. Interpolation representation of feedforward neural networks. *Mathematical and Computer Modelling* 37, 829–847.
- Mace, S., Mata-Alvarez, J., 2002. Utilization of SBR technology for wastewater treatment: an overview. *Industrial and Engineering Chemistry* 41, 5539–5553.
- Marsili-Libelli, S., 2006. Control of SBR switching by fuzzy pattern recognition. *Water Research* 40, 1095–1107.
- Marsili-Libelli, S., Spagni, A., Susini, R., 2008. Intelligent monitoring system for long-term control of sequencing batch reactors. *Water Science and Technology* 57 (3), 431–438.
- Olsson, G., Newell, B., 1999. *Wastewater Treatment Systems: Modeling, Diagnosis and Control*. IWA Publishing, London, UK.

- Olsson, G., 2006. Instrumentation, control and automation in the water industry – state-of-the-art and new challenges. *Water Science and Technology* 53 (4–5), 1–16.
- Olsson, G., Nielsen, M.K., Yuan, Z., Lynggaard-Jensen, A., Steyer, J.P., 2005. Instrumentation, Control and Automation in Wastewater Treatment Systems. Scientific and Technical Report No. 15. IWA Publishing, London, UK.
- Poch, M., Comas, J., Rodríguez-Roda, I., Sánchez-Marré, M., Cortes, U., 2004. Designing and building real environmental decision support systems. *Environmental Modelling and Software* 19 (9), 857–873.
- Rodríguez-Roda, I., Sánchez-Marré, M., Comas, J., Baeza, J., Colprim, J., Lafuente, J., Cortes, U., Poch, M., 2002. A hybrid supervisory system to support wastewater treatment plant operations. *Water Science and Technology* 45 (4/5), 289–297.
- Sánchez-Marré, M., Cortés, U., Martínez, M., Comas, J., Rodríguez-Roda, I., 2005. An Approach for Temporal Case-based Reasoning: Episode-based Reasoning. *Lecture Notes in Computer Science, Case-based Reasoning Research and Development*, vol. 3620. Springer-Verlag, Berlin Heidelberg, 465–476.
- Sottara, D., Colombini, G., Luccarini, L., Mello, P., 2009. A pool of experts to evaluate the evolution of biological processes in SBR plants. In: *Proceedings of HAIS09, Fourth International Conference on Hybrid Artificial Intelligence Systems*, Salamanca, Spain, June 10–12, 2009, springer.
- Spagni, A., Buday, J., Ratini, P., Bortone, G., 2001. Experimental considerations on monitoring ORP, pH, conductivity and dissolved oxygen in nitrogen and phosphorus biological removal processes. *Water Science and Technology* 43 (11), 197–204.
- Van der Aalst, W.M.P., van Dongen, B.F., Gunther, C.W., Mans, R.S., de Medeiros, A.A., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M., 2007. ProM 4.0: comprehensive support for real process analysis. In: Kleijn, J., Yakovle, A. (Eds.), *Application and Theory of Petri Nets and Other Models of Concurrency*, vol. 4546, pp. 484–494 (ICATPN 2007).
- Van Dongen, B.F., van der Aalst, W.M.P., 2005. A meta model for process mining data. In: Casto, J., Teniente, E. (Eds.), *Proceedings of the CAiSE'05 Workshops*, vol. 2. FEUP, Porto, Portugal, pp. 309–320 (EMOI-INTEROP Workshop).
- Water Environment Federation, 2006. *Automation of Wastewater Treatment Facilities*. WEF Manual of Practice n. 21, third ed. WEF Press and McGraw-Hill, Alexandria Virginia.
- Weske, M., 2007. *Business process management: concepts, languages, architectures*. Springer-Verlag Berlin Heidelberg, ISBN: 978-3-540-73521-2.
- Yoo, C.K., Lee, D.S., Vanrolleghem, P.A., 2004. Application of multiway ICA for on-line process monitoring of a sequencing batch reactor. *Water Research* 38, 1715–1732.
- Zhao, H., Hao, O.J., McAvoy, T.J., 1999. Approaches to modelling nutrient dynamics: ASM2, simplified model and neural nets. *Water Science and Technology* 39 (1), 227–234.