A software platform for real-time control and monitoring of a wastewater treatment plant
A. Sánchez, M. Wade and M. R. Katebi
Transactions of the Institute of Measurement and Control 2005; 27; 153
DOI: 10.1191/0142331205tm142oa

The online version of this article can be found at: http://tim.sagepub.com/cgi/content/abstract/27/3/153
A software platform for real-time control and monitoring of a wastewater treatment plant

A. Sánchez*, M. Wade and M.R. Katebi

Industrial Control Centre, University of Strathclyde, Glasgow G1 1QE, UK

This paper reports on the design, implementation and real-time operation of advanced process control and monitoring techniques for wastewater treatment plants. The paper presents the development of a software platform and its implementation in a full-scale wastewater treatment plant. The software platform allows the real-time execution of advanced process control techniques as model predictive control, subspace identification, and statistical process analysis using principal component analysis methods. Results obtained by the real-time execution of such algorithms are also presented.

Key words: model predictive control; real-time control; subspace identification; wastewater treatment.

1. Introduction

In the case of the wastewater industry, the introduction of SCADA (supervisory control and data acquisition) systems has facilitated the control and monitoring of processes with several hundreds of control loops. However, it is only in the last decade or so, with effluent quality standards becoming increasingly stringent, that large-scale urban wastewater treatment plants (WWTPs) have been equipped with similar supervisory control systems. Some of the principal characteristics of these systems are the human–machine interface (HMI), which allows the operator to have a plant-wide perspective of the operation of the plant and large databases to collect important data.
plant information such as alarms, events and process variables. However, due to security and time constraints, control actions in SCADA systems are often limited to very simple functions such as pump and valve scheduling or modifying some parameter settings as setpoints, with the more sophisticated control algorithms being in the field in devices such as programmable logic controllers (PLCs) or temperature controllers (Katebi et al., 1999).

Time constraints appear because of the necessity of maintaining real-time operation. In most cases, communication between the high-level control and low-level hardware is slow and performed by polling due to the high channel density so direct real-time control is difficult and unreliable from this point of view. Security issues appear when the upper-level control operates over unreliable operating systems. In some cases, where security is life critical, upper-level control systems operate over industrial hardware and software, which improves reliability and reduces long-term costs.

In the case of the wastewater treatment industry, many of the processes and variables are very slow and high-level security is not imperative. However, since in activated sludge WWTPs plants, biological processes and chemical reactions are involved, the system performance and operation is susceptible to external disturbances. Disturbances can come in the form of increased influent flow, and nutrient or chemical loading due to weather conditions or industry discharges. Erroneous handling of the plant can produce total inhibition or even death of the plant biological components and therefore halt operation for weeks or even months until sludge inventories are recovered. Therefore in these types of processes, it is very important to have plant-wide information in order to assess the operation and take appropriate control actions to avoid plant mismanagement.

This paper presents the development and implementation of advanced process control and process monitoring in real-time. Much of the content of the paper is concentrated in the development of a software platform to perform these functions and its interfacing with the industrial plant. The software has been developed employing two commercial software packages: MATLAB with SIMULINK (Mathworks) and LabVIEW with the Datalogging and Supervisory Control module plus some additional toolkits (National Instruments). Matlab has been employed to write the code of many of the algorithms and controllers. This code has been compiled into dynamic link libraries (DLLs) and included in the LabVIEW schematics. The interfaces with the PLC and the user, have been programmed in LabVIEW. The paper also presents results of the execution in real-time of these control algorithms in a full-scale WWTP in Scotland.

The presentation of the results has been divided in two papers. Part 1, which is presented in this paper, begins with a brief description of the WWTP where the full-scale implementation has taken place. Later, in section 3 the development and implementation of the software is described. The section introduces the reader into some of the basics of the software and the hardware integration and the communication protocols and interfaces. Sections 4 and 5 present results obtained by executing the algorithms in real-time in the plant, for identification and control respectively. Finally, the paper ends with a summary of the main achievements and conclusions.
The contents of Part 2 is focused on algorithms for process monitoring employing algorithms for statistical process analysis.

2. Swinstie WWTP

Swinstie WWTP is located close to the town of Cleland, about 25 km south-east of the City of Glasgow in the UK. The plant was commissioned in 1998 and has a capacity of 20,075 p.e. (population equivalents) but the actual load on the plant is typically 13,000 p.e. The influent wastewater is predominantly household effluent and very little is industrial discharge.

Following a process review by Scottish Water in May 2003 a revised plant configuration was implemented for Swinstie WWTP due to over sizing of the main unit operations at the plant. The details of the actual plant configuration are provided below.

2.1 Sewer network

The sewer network serves a number of small communities close to Swinstie and the main trunk sewer enters the plant by gravity alone. At present, there is no control on the volume of influent to the plant, it passes immediately into a channel that incorporates a control flume to limit the flow to the works in periods of high hydraulic loading.

2.2 Plant configuration

The plant inlet has no control mechanism for regulating and distributing the influent. All flows in excess of four daily dry weather flow (DDWF) bypass the plant and are discharged into the River Calder at the south side of the site after passing through a 6-mm mechanical screen.

Preliminary treatment consists of 6-mm screening provided by duty standby mechanical raked screens. A hand-raked emergency bypass screen is also provided.

Grit removal is provided by two detritors downstream of the screens. There is no fat or grease removal in this unit due to the low amounts observed in the influent, although the primary tanks do have scum traps that remove the small amount of fatty material and debris that passes to the primary treatment stage.

Primary treatment consists of two primary settling tanks. Excess return activated sludge (ERAS) is returned to the primary tanks for cosettling. Scum and settled solids are removed to a sludge holding tank and the wastewater is passed to a pretreatment channel prior to the secondary treatment stage.

A flume in the channel leading to secondary treatment limits the flow to full treatment to two DDWF. Settled sewage in excess is deviated to the river via 6-mm mechanical screens. The aeration is preceded with an anoxic zone (one anoxic tank) for denitrification and sludge conditioning. Secondary treatment consists of two parallel aeration tanks. Aeration is provided by fine bubble diffused aerators (FBDA) situated...
in three zones down the length of each tank. This provides a stepped oxygen supply along the length of the aeration tank.

Mixed liquor from the aeration tank is dosed with ferric sulphate to remove phosphate prior to final clarification. An internal recycle facility is also provided within the aeration tanks.

Final clarification is provided by two final settlement tanks with equal distribution of the influent. A proportion of the settled solids are returned to the secondary treatment stage as return activated sludge (RAS). The clarified water passes through a micro-strainer for polishing before discharge to the river. The residue is returned to the primary settling tanks for cosettling. All sludge is removed by tanker and treated off-site. Improvement of the control system relies on knowledge of the existing structure of the wastewater systems. Figure 1 presents the layout configuration of Swinsite WWTP.

2.3 Instrumentation and control

The current available data acquisition and control system at Swinstie consists of an IC2000 SCADA unit with process and measurement visualization capacity.

The works are controlled by a Siemens TI-565 PLC located in the power distribution house. The system contains all the algorithms for the control of the plant.

The SCADA software provides a limited amount of control by allowing setpoints to be altered and communicates with the user via mimic screens and associated alarms. The system has one workstation for user input located in the main control room. This comprises a VDU, keyboard, mouse and two printers.

Figure 1 Swinstie WWTP plant layout

Downloaded from http://tim.sagepub.com at Univ of Newcastle upon Tyne on February 10, 2010
There is no control provided except that the setpoints for the controlling parameters, e.g., desludging times, detritor operating times, dissolved oxygen (DO) levels, can be altered within design set limits.

The aeration control system employs four measurements of DO, two in each lane (inlet and outlet). The measurements are averaged and this value is compared with a high and low DO range. If the mean is above the range, the controller will decrease the blower speed by 10% every 1 min for the high-capacity blowers and 10 min for the low-capacity blowers. Similarly, a 10% speed increase will occur if the DO mean is below the established range. This is not an ideal control system, as there is no account for variation in DO between lanes and percentage speed variation is a very imprecise form of control.

A certain amount of aeration is required to maintain an appropriate mixture and preventing the blowers stalling. This, however, results in a very limited range of speed to which the blowers can actuate to perform control actions.

The anoxic zone has one DO measurement that is controlled independently. When the DO is above a set limit, the output penstocks to the aeration tanks will close fractionally, which will increase the hydraulic retention time and mean cell residence time in the anoxic zone, thus reducing the DO in this section.

ERAS is controlled in a simple manner. When the level of ERAS in the chamber reaches a set level, the pumps switch on and transport the ERAS to the primary settlement tanks until the ERAS level reduces to a set minimum.

There is currently no control structure in place for the RAS flow, although the screw pumps can be switched on or off depending on requirements.

3. Software development

The wastewater process industry has been the focus of intensive research and development in the last 20 years. The state-of-the-art in sensor technology to acquire online measurements of nutrients, e.g., ammonia, nitrate, phosphate, and physical variables such as DO or suspended solids, has reached a state in which advanced data quality management and control strategies can be used to improve process performance. It is claimed in Yuan et al. (2001) that innovative process design and optimized process control is the solution to many of the problems involving nutrient removal. However, the use of sophisticated control algorithms and data quality management is still not widely used. For example, as concluded by Yuan et al. (2001), the use of model-based online control is still in its infancy, and it is just in the past 15 years that online monitoring and data acquisition has helped to generate and verify mathematical process models which can be used for control. Data monitoring and validation is an important area of research. Its use has helped to improve plant performance by improving plant design and operation. Many new approaches such as Aeration Tank Settling (ATS; Nielsen et al., 2000) or step-feed to increase hydraulic capacity to cope with large weather disturbances have arisen from detailed data monitoring and analysis.

Research in integrated wastewater treatment systems (sewer and treatment plant) are also now of interest (Nielsen et al., 1996), and data monitoring plays a very
important role, since by analysing plant data it is possible, for example, to identify influent flow models (Nielsen, 2002, SMAC project report). Plant monitoring can also be used for knowledge extraction, which allows the assessment of plant operation by identifying deviations from the normal operation conditions (Nielsen, 2002, SMAC project report).

In this context, a supervisory control system for a WWTP would not only allow the collection of data and its analysis but also allow the use of the acquired knowledge to improve plant performance. It would be expected that under this scheme it will be possible to implement more sophisticated control algorithms and not only be limited to conventional SCADA control functions. The platform introduced in this paper provides an alternative solution to the lack of versatility for computational requirements in existing SCADA systems. The platform has tools that can be interfaced with an existing SCADA or used independently, since for security reasons a plant can usually operate without the SCADA. This solution can only be implemented when the plant time constants of the specific loops are larger than the polling time of the existing SCADA, as is the case of the activated sludge wastewater treatment process. The platform is conceived to work in parallel with the existing SCADA. However, this might not always be possible, depending on the functionalities of the existing supervisory system.

3.1 Software platform architecture

The platform architecture is comprised of three functional units, which are linked to the plant SCADA system or directly with the PLC, which in turn communicates with the SCADA as shown in Figure 2. The three units in the control platform are the Process Control unit that monitors and tunes the control loops, the HMI unit, which is an interactive tool for visualization of information collected in the system, and the Data Quality Assessment (DQA) unit, which performs the process monitoring and knowledge extraction tasks.

![Platform architecture diagram](http://tim.sagepub.com)
The PLC link employs an OLE (object linking and embedding) for Process Control Server (OPC). OPC servers are commercially available and are usually supplied by the PLC manufacturer.

The following sections give a more detailed explanation of each of the units.

3.1.1 Process control unit: The process control unit provides various functions to perform process identification and control. For the process identification the implemented functions perform the following tasks,

- *Experiment design and real-time execution*: Subspace identification requires that the data employed show the system persistently excited. In order to obtain such data, it is necessary to excite the system by performing an experiment. The design of the experiment consists of selecting the control variables of interest to which probing signals will be applied. A pseudo-random binary signal (PRBS) generator has been implemented as a probing signal, which can be modified in amplitude, mean and time-step. The time-step is selected according to the response time of the process to be identified. Fast variables require a faster time-step, whereas slow variables require a higher time-step. In the activated sludge process, oxygen is a fast variable, which ranges in the minute time horizon, therefore a time step of about 2–3 minutes is usually adequate. The module also allows to simulate the probing signals in real-time before applying them to the process. With the probing signals ready, the experiment can be run in real-time. The software database will automatically record all the signals (inputs and outputs). Once sufficient data has been collected, the data is exported from the database into a normal text file. This is not an automatic process, since there might be the need to select only a subset of the recorded data. Usually these experiments can take up to 15 h in fast variables as oxygen, time during which many unexpected events can occur. Therefore the need for manually selecting appropriate data.

- *Subspace identification*: Two subspace identification methods have been implemented: 1) robust N4SID ‘SV’ and 2) robust N4SID ‘CVA’ (van Overschee and De Moor, 1996). Both algorithms have been programmed in MATLAB and compiled into C shared libraries (DLLs). The original implementation code of the two subspace identification algorithms is from the toolbox provided in (van Overschee and De Moor, 1996) and freely available in the following ftp server: ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/vanoverschee/book/subfun/. Some modifications to the code have been necessary in order to correctly interface the algorithm into LabVIEW. The core of the implementation of this module consists of two software blocks executed in sequence, as presented in Figure 3. The blocks orderdll.dll and idendll.dll are dynamic link libraries (DLL) compiled from MATLAB.
The first library receives data read from the text file, and performs the decomposition step necessary in subspace identification algorithms. The library outputs are the singular values if the selected algorithm is SV, and the principal angles if the selected algorithm is CVA. The LabVIEW implementation then gives a graphical representation of the singular values or the principal angles. The user is then able to estimate the order of the model to identify. This is passed into the second DLL which calculates the model.

- **Model analysis and simulation**: Once the order is selected and the model identified, the module will give an indication concerning the stability. The three possible indications are: 1) stable, 2) unstable and 3) marginally stable. The model simulation is performed using the same data used for the identification. The response of the model is presented in a graph, which also includes the input and output signals used in the identification. These functions are totally implemented using LabVIEW function blocks.

Figure 4 presents the graphical user frontend of this unit.

The functions related to process control make use of the models identified and analysed using the identification tools. The control functions provide algorithms to design and run a hierarchical model predictive controller (MPC) in real-time. These functions comprise the design of the MPC controller given a model identified using the identification functions. These functions are:
• **Observer design**: The observer is designed by pole placement, and is implemented using the MATLAB script node. Also an output disturbance model is employed to account for plant–model mismatch.

• **Predictor design**: The predictor module calculates the matrices $\psi$, $\Gamma$ and $\Theta$ of the standard MPC formulation presented in Maciejowski (2001).

• **Constraint specifications**: The module also allows the inclusion of constraints if required. These modules are implemented using a DLL compiled from MATLAB.

• **MPC controller execution control and monitoring**: Once the observer and the predictor have been designed, the unit allows the real-time execution of the controller in the plant. During the real-time execution, the system allows the fine tuning of the MPC controller, by modifying horizons, and weights. The numerical implementation of the online optimization employs a least-square approach as formulated in Maciejowski (2001).

Figure 5 shows the user graphical frontend of the MPC unit.

3.1.2 **DQA unit**: This unit monitors the process state in real time using online measurements and historical information stored in the database. By monitoring the state of the plant in real-time it is possible to detect faults and even provide a diagnosis. Process monitoring for fault detection and diagnosis is a complex area of study with continuous developments (e.g., Wade, 2004).
The unit retrieves data from the database and online measurements, and processes it with the purpose of extracting knowledge of the process behaviour and plant operation. Process knowledge extraction accumulates information from expert knowledge of the plant (plant operator), online data and historical data, to find possible ways of globally optimizing the plant by a better scheduling of actuators or by enhancing setpoints by identifying current or future operating states. It also provides means by which abnormal plant conditions can be predicted and alarms rose. Therefore this unit comprises software algorithms to perform the following tasks:

- **Recursive statistical process monitoring**: Recursive statistical process monitoring employs a recursive formulation of principal component analysis (PCA). PCA is a multivariate statistical tool, which allows the comprehension of data and examination of its statistical properties, employing geometrical tools. The recursive principal component method employed in the software is the algorithm proposed by Li et al. (2000) for the recursive update of the correlation matrix $R_k$ at each sampling time. Using the recursive update of the correlation matrix, it is possible to perform the following: 1) statistics monitoring, by calculating the $Q$ and Hotelling’s $T^2$ statistics of the compressed data set (Chiang et al., 2001); 2) fault detection, by comparing the statistics with limits within a 95 and 99 percentile confidence interval; and 3) diagnosis, by observing the loads and scores from the data set decomposition. The recursive update of the correlation matrix has been implemented in Matlab and is passed into LabVIEW by using a script node. Figure 6 shows a block diagram of the execution of the algorithm.

The first step is to feed data into an initial data block set, by looking into the database and selecting appropriate data. This data must representative of the process operation and must not contain important failures in the process operation. Once the variables and the data have been retrieved and fed into the initial data-block set, the recursive update of the correlation matrix begins.

Once the correlation matrix $R_k$ and a normalized data block matrix $X_k$ are calculated, they are used in the fault diagnosis module to identify the system state and detect where and what type of fault is occurring.

- **Fault detection and diagnosis**: Fault detection is performed by calculating Hotelling’s $T^2$ statistic and the $Q$ statistic using the normalized data matrix $X_k$ and the correlation matrix $R_k$ (see Chiang et al., 2001, for a definition of the statistics). The calculation of these statistics is performed by decomposing the correlation matrix. Two decomposition methods have been implemented: 1) singular value decomposition (SVD) 2) Lanczos tridiagonalization. The choice of the decomposition matrix depends on the amount of data and type of process. Lanczos tridiagonalization allows the the SVD decomposition in large sparse data matrices. The reason for implementing both

![Figure 6 RPCA execution](http://tim.sagepub.com)

”Software platform for wastewater treatment plant”
methods is to investigate their suitability for the activated sludge process. The Lanczos algorithm has been implemented using a freeware toolbox for MATLAB called LANPRO ver. 1.0 available at (http://soi.stanford.edu/~rmunk/PROPACK/index.html). The detection of faults is performed by comparing the value of the statistics with a confidence interval of 95 and 99 percentile. The decomposition of the correlation matrix also provides the loads and scores. The examination of these vectors allows the recovery of information regarding which measurement in the process is contributing more extensively to the fault. Even more, by using more advanced algorithms not implemented in this software, it is possible to identify the type of fault.

A more detailed explanation of this unit and its application in the Swinstie WWTP is reported in Wade et al. (2004). Finally, Figure 7 gives a view of the graphical user frontend of the unit.

3.1.3 The HMI unit: Even though, in itself the HMI is not something really new, the original contribution made by this reported research is to synthesize all the complex information from the data assessment and control units in a way that can be readily understood by wastewater technical staff, staff who may not be control engineering experts. Figure 8 shows a schematic from an HMI designed specifically for the test plant at Scottish Water.

Figure 7 User interface of the data quality assessment unit
3.2 Software platform implementation

Due to the large number of commercially available SCADA systems in the market, it is nowadays very difficult to implement a system that is compatible with purchased proprietary software. The main problem within the architecture of the system is to interface in an adequate manner with the existing software and hardware.

Some SCADAs are designed in such a way that they only provides communication protocols with software from the same manufacturer. Even if the manufacturer provides some type of well-known industry standard communication protocol, the allowed functions are usually very limited.

The platform developed employs OPC technology to interface with the PLC. OLE is a Microsoft component which allows the automation of certain processes and applications like communication servers. OPC was designed for industrial application purposes.

The software platform interfaces with the PLC by registering with an OPC server which handles the communication flow. Each type of PLC requires a particular OPC server, which is usually provided by the manufacturer. In the case of Swinstie WWTP, the PLC is a TI 565 (Siemens), which communicates using a Tiway protocol. Therefore the selected OPC server should support this protocol and the communication interface, which in the case of Swinstie is an RS-232. If a network were present, then the interface would be through an ethernet link in most cases.
With the OPC server correctly configured and running, the next step is to populate the database. PLCs contain memory registers from where they read or save information that is to be used by the internal program. In general, each input–output of the PLC is also mapped into a memory location (or register). The access to these memory locations is immediate using the OPC server; however, it is important to map it into the database, so the information read or written to the register is mirrored in the client (platform database). LabVIEW and the DSC module employ a tag engine concept. A tag is a variable associated with a register; however, it is already located in the client and not in the PLC, therefore providing faster access. The tag engine ensures that the register is accessed at the required sampling times or when the program requires. This helps the system to have a deterministic access to the variables of the process, and therefore perform operations in real-time.

Concerning the implementation of the different algorithms, some were programmed over MATLAB and compiled into C shared libraries (DLLs) using the MATLAB compiler and some additional C code to interface with common C standards and not particularly to MATLAB. All of the routines have been tested in simulation exercises to ensure their correct execution.

Other modules have been implemented in SIMULINK and integrated into LabVIEW employing the Simulation Interface toolkit. Table 1 gives a summary of the technologies employed to implement different algorithms from MATLAB into LabVIEW.

### 4. Real-time identification of DO

This section shows the results obtained by the real-time execution of the identification module. Some of the results obtained demonstrate that DO can be modelled by low-order models, and that this model can be used in a MPC.

Several identification runs where performed at Swinstie, by exciting the plant with a PRBS signal to identify a closed-loop model. To do so, several limitations had to be overcome to make the experiments possible. First of all, the aeration capacity in Swinstie is over designed. The aeration system is composed of four blowers: two of low capacity and two of high capacity. The plant normally requires a small lower to pump sufficient air into the system. However, since the plant is configured to operate only at half its design load, even the small blower at its slowest speed over-aerates the

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Technology</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPCA</td>
<td>MATLAB script node (Activex)</td>
<td>Process monitoring</td>
</tr>
<tr>
<td>PRBS</td>
<td>Simulation Interface Toolkit (DLL)</td>
<td>Identification</td>
</tr>
<tr>
<td>Subspace Id.</td>
<td>MATLAB Compiler (DLL)</td>
<td>Identification</td>
</tr>
<tr>
<td>Observer design</td>
<td>MATLAB script node (Activex)</td>
<td>Control</td>
</tr>
<tr>
<td>MPC design</td>
<td>MATLAB Compiler (DLL)</td>
<td>Control</td>
</tr>
</tbody>
</table>
plant. The effect of this over-aeration is that the DO levels are almost always over the
specified range of 1.8–2.2 mg/l.

A second limitation in the blower actuator system has to do with the blower update
speed. The small blowers update their speed (increase or decrease) at a rate of 10 min,
and the big blowers at a rate of 1 min.

Because of these two limitations, the only possible solution that did not demand the
high cost of PLC re-programming was to perform all the experiments at a high DO
level using a big blower. This approach, however, does not invalidate the results, since
as proposed in papers before and as will be demonstrated in the following section, DO
can be modelled accurately with low-order linear models within almost the whole
range. Additionally, to reduce the effect of the dead-band introduced by the control
algorithm programmed in the PLC, the high–low range has been reduced to a
minimum of 1 mg/l. The subspace identification methods used were robust-N4SID SV
and CVA.

The results of these tests lead to the second-order linear model of Equations (1)
and (2).

\[
x(k + 1) = \begin{bmatrix} 0.97788 & -0.04357 \\ 0.03658 & 0.84292 \end{bmatrix} x(k) + \begin{bmatrix} -0.04629 \\ 0.1124 \end{bmatrix} r(k)
\]  

(1)

\[
y(k) = \begin{bmatrix} -0.38536 & -0.18981 \end{bmatrix} x(k) + \begin{bmatrix} -0.0067 \end{bmatrix} r(k)
\]  

(2)

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure9}
\caption{Identification experiment in Swinstie WWTP}
\end{figure}
Figure 9 shows the trends used for the identification. Notice that just after sample 800, there is a switch between blowers, and the update time of the smaller blower produces a significant change of behaviour in the system response. Because of this, this part of the data set is not to be used in the identification. Figure 10 shows the response of the identified model compared to the actual system response, where the high–low range has been averaged for better illustration. Notice there is an offset between the signals; thus a disturbance model should be used to correct the offset when designing the predictor.

5. MPC control of DO in Swinstie WWTP

The design and evaluation of a MPC controller for DO in Swinstie WWTP is presented in this section. The MPC controller is designed using the model identified in the previous section. The controller uses the state-space formulation reviewed in Maciejowski (2001). The obtained results show a degree of improvement compared with the controller implemented in the plant PLC. The section is organized in such a way that the design of the MPC controller is presented first, followed by results showing the improvement gained using the MPC controller.
5.1 MPC design

The first component to be designed for the MPC is the observer. A constant output correction has been introduced to compensate for unmeasured disturbances and plant–model mismatch. The observer was designed by placing the closed-loop poles as follows,

\[
Poles = [0.7 \quad 0.6 \quad 0.5]^T
\]

Figure 11 shows the observer response when initialized. It can be seen that the observer requires around 25 min to converge to the actual measurement. Also, due to its large overshoot, it is of vital importance to allow sufficient time for the observer to converge before activating the MPC controller. This is just a precaution to avoid the controller producing unnecessary large control actions. Figure 12 presents an interesting result regarding the linearity of DO. Notice in this figure that even though the oxygen level drops sharply, the observer is still able to track reasonably well the trajectory.

The predictor and optimization parameters are presented in Table 2. These values have been obtained by carefully tuning the controller in real-time operation.

![Figure 11 DO observer response in Swinstie WWTP](http://tim.sagepub.com)

**Figure 11** DO observer response in Swinstie WWTP
One of the main advantages of using a predictive controller is the facility to include constraints in the optimization process. It is in this way that the physical limitations arising from the actuators (air compressors) are included when solving the optimization. Constraints also allow the inclusion of operation conditions that are necessary for the process to work. Thus constraints allow limits to be imposed over variables which in practice cannot go under or over certain limits, as for example the oxygen concentration cannot be less than zero.

The optimization algorithm employed in the implementation of the MPC controller is a constrained least-squares. The use of this algorithm to solve the QP (quadratic programming) problem guarantees a robust numerical implementation. The implemented code also contains a protection mechanisms for infeasible problems, by using the last known optimization solution.

### Table 2 Prediction and optimization parameters

<table>
<thead>
<tr>
<th>$H_p$</th>
<th>$H_u$</th>
<th>$Q$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>20</td>
<td>3</td>
<td>0.5</td>
</tr>
</tbody>
</table>
5.2 DO control

The MPC controller implemented in Swinstie WWTP presents some degree of improvement over the existing controller. More tests are required in order to tune the system adequately and obtain a significant benefit. In general, the use of the MPC controller in Swinstie is limited to the plant actuators, since there is no possibility of performing this control at acceptable oxygen levels of around 2–3 mg/l. This limitation arises from the limited speed range of the blowers and the absence of a more sophisticated control structure (e.g., pressure loop, and motorized valves). However, just for the purpose of demonstration, the MPC has been set to run with high DO setpoints.

Figure 13 shows a step response of the system operating with the MPC. Notice as well in this figure, that the MPC is activated only after the observer has converged. Figure 14 shows a comparison between the control algorithm in the PLC operating by its own, and the system operating with the MPC.

6. Conclusions

This paper has presented the development and implementation of a control software platform, by the authors, for advanced process control and process monitoring. The
development of the software has used several technologies to implement the different algorithms. MATLAB and LabVIEW have been employed to develop the software. The integration of code with the two softwares has been possible using: DLLs and Activex. The integration of MATLAB programmed algorithms into LabVIEW has been performed using several technologies so the code could be executed in real-time. A disadvantage of this approach are that MATLAB requires additional high-level programming (C), so it can be interfaced by standard C types. The code execution is, however, much faster than in the MATLAB environment.

The interfacing of the software with the PLC in the plant has been implemented using OPC technology. OPC technology allows a fast and efficient communication with the PLC.

The platform has been succesfully tested in full scale in Swinstie WWTP. The tests performed in real-time are: 1) subspace identification, 2) MPC controller design, and 3) process monitoring. The tests results have also corroborated previous results obtained by simulation (Sánchez and Katebi, 2003). Some of the most important convey the linearity of DO under the saturation level therefore allowing the assumption of simple first- or second-order model approximations consistent with the rest of the work presented in this paper.

Figure 14 Variance reduction when using a MPC for DO control
References


