
M.J. Wade*,†, R. Katebi and A. Sanchez

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

This paper reports on the design, implementation and real-time operation of advanced process 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 monitoring techniques that are used for fault and process upset detection and identification. The statistical methods are part of a supervisory control level that allows for integration of the different facets of process monitoring and control, including fault detection, knowledge extraction and controller action. Results obtained by the real-time execution of such algorithms are presented.

Key words: Wastewater Treatment, Real-time Control, Process Monitoring, Recursive Principal Component Analysis

List of Symbols

\[ a = \text{Selected number of Principal Components;} \]
\[ E = \text{Residual Matrix;} \]
\[ l = \text{Principal Components;} \]
\[ m = \text{Number of columns;} \]

*Email: m.wade@strath.ac.uk, Tel: +44 (0)141 5482073
†Current Address: 6.14C, Royal College Building, Dept. Pure & Applied Chemistry, University of Strathclyde, 295, Cathedral Street, Glasgow, UK G1 1XL
\( n = \) Number of rows;
\( Q = Q\)-residual statistic;
\( T = \) Loading Matrix;
\( T^* = \) Hotelling’s statistic;
\( U = \) Orthogonal Matrix;
\( v = \) Score vector;
\( W = \) Orthonormal vector;
\( x = \) Data vector;
\( \Sigma = \) Singular Value Matrix;
\( \Lambda = \) Eigenvalue Matrix.

To avoid duplication, versions of upper-case symbols indicate matrices and lower-case symbols indicate vectors corresponding to those listed here. The superscript \( T \) is indicative of the transpose of a vector or matrix.

## 1 Introduction

Methods for improved process monitoring and knowledge discovery have been applied to wastewater treatment plant (WWTP) measurement data, (Savic et al., 1999; Rosén & Lennox, 2001). However, the tools are limited to theoretical analysis or off-line demonstrability, with no indication of their robustness and practicality for real-time, on-line applications. Because models are only approximate representations of the real system or process, simulation or numerical analysis cannot truly capture the behaviour or response of the system in actuality. Likewise, tools developed from mathematical principles and tested using model-based or data-driven techniques may not have real applicative value due to the limits of the model or the quality of the data, respectively.

Here, tools based on Multivariate Statistical (MVS) techniques are implemented in a real-time environment and executed on-line for process monitoring and as part of the control system of Swinstie WWTP, Glasgow, UK. The transition from off-line to on-line working mode is achieved by using a software platform based on LabView (National Instruments, Austin, Texas) and its communication tools. Furthermore, the on-line functions are converted to stand-alone applications to remove the problems with running multiple software programmes simultaneously and are developed using open source code. The stand-alone applications are a self-contained package of unsupervised functions hidden behind a user-friendly front end that can be used as an interactive module or for supervision purposes.
The paper is organised in four parts. The first part is an overview of Principal Component Analysis, which forms the basis of the algorithms used for fault detection and identification. There then follows a discussion on the practical implementation of the process monitoring tools at Swinstie WWTP using the LabView software. Next a discussion of the supervisory control structure for the treatment plant is provided to highlight the possibility for integration between the separate process monitoring and process control facets of plant operation and assessment. Finally, some results are presented for on-line fault and process monitoring in real-time at Swinstie WWTP, which show the capabilities of the MVS tools for handling data as it becomes available.

2 Overview of Principal Component Analysis

The fundamental concept of PCA is to transform a large set of data containing \( m \) associated variables \( \{x_m\} \) into a smaller set of uncorrelated variables \( \{x_l\} \), termed Principal Components (PC), whilst maintaining the largest amount of information relating to the variation between the variables in the original data set. In this way, previously indistinguishable patterns or latent information may be extracted from large amounts of process data. Furthermore, the dimensionally reduced variable space can be used for monitoring subsequent changes in the variance between variables as new data becomes available.

The \( j^{th} \) principal component \( l_j \) can be expressed as the linear combination of the measured variables, \( x \) and associated weighting factors (loadings), \( v \):

\[
l_j = v_{j1}x_1 + v_{j2}x_2 + \ldots + v_{jm}x_m
\]

(1)

which can be reduced to:

\[
l_j = v_j^T x
\]

(2)

where \( v_j^T \) is a vector containing all the \( j^{th} \) loadings and \( l_j \) has the greatest variance subject to two conditions:

\[v_j^T v_j = 1\]

\[v_j^T v_i = 0 \quad (i < j)\]
Denoting $V \in \mathbb{R}^{n \times a}$ as the matrix of loading vectors associated with the first $a$ principal components and $X \in \mathbb{R}^{n \times m}$ as the original data matrix (training set), then the projections of the $n$ samples (observations) in $X$ to the lower-dimensional space are captured in the scores matrix:

$$T = XV$$

such that $T \in \mathbb{R}^{n \times k}$. Projecting the score matrix back into the sample (observational) space yields:

$$\hat{X} = TV^T$$

It can be seen that some loss of information occurs when $a < m$, so that:

$$X = TV^T + E$$

where $E$ is the residual matrix, which captures the variations in the sample space not described by $\hat{X}$, i.e. the $m - a$ smallest PCs. In general, the residual matrix will contain information pertaining to the random variations in the original data, i.e. it has a small signal to noise ratio, (Chiang et al., 2001). The removal of $E$ from the sample space will result in the generation of a better process model description in which the important variations may be analysed. If $a = m$ then $E = 0$ and all the information from the original data matrix is retained and all the variability directions will be described. However, with $a < m$, then it is possible to partition the sample space into a process space (the PCA model) and a noise space (the residual matrix) as shown in Fig. 1.

Hence, when performing PCA, it is imperative to select the number of principal components ($a$) that capture the important variations in the data without retaining the random effects present in the noise space. In effect, PCA can be viewed as a noise filtering technique whose effectiveness depends on optimally selecting the number of PCs to retain in the model.

The Singular Value Decomposition (SVD) technique is used for determining the loading vectors by solving the stationary points (singular values) of a matrix that is non-square, so that:

$$X = U \Sigma V^T$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{m \times m}$ are orthonormal and unitary matrices and $\Sigma$ is a diagonal
matrix containing all the non-negative real singular values ($\sigma$) with decreasing magnitude along the main diagonal:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$$

The loading vectors ($v_m$) are the orthonormal column vectors of $V$. The SVD method enables the PCs to be found in a computationally efficient manner as well as the PC scores and, hence, some ability to graphically represent the PCA results, Jolliffe (1986), i.e. scores and loadings plots.

The SVD method can be transformed into an eigenvalue problem by some simple matrix theory. Given $X^TX \in \mathbb{R}^{m \times m}$ is a symmetric matrix, then the eigenvalue decomposition (ED) of this matrix results in, (Gonnet, 2002):

$$X^TX = WW^T \quad (7)$$

where $W \in \mathbb{R}^{m \times m}$ is orthonormal and $\Lambda$ is a diagonal matrix with non-negative real eigenvalues ($\lambda$) with decreasing magnitude along the main diagonal:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min(m,n)} \geq 0$$

Applying equation (7) to equation (6) yields:

$$X^TX = V\Sigma^TU^T U\Sigma V^T \quad (8)$$

Since $U^TU = I$, then equation (8) becomes:

$$X^TX = V\Sigma^T \Sigma V^T \quad (9)$$

and, as $\Sigma^T \Sigma \in \mathbb{R}^{m \times m}$ contains the diagonal elements $\sigma^2$, it can be concluded that:

$$X^TX = V\Lambda V^T \quad (10)$$

which is the covariance matrix of the of the original data set, where $\lambda_i = \sigma_i^2$ for $i = 1, \ldots, m$.

Often the training data ($X$) is normalised before PCA is performed, particularly if the measurement vectors (columns of the matrix) are of different orders of magnitude due to mismatch in
units, for example. The autoscaling of the matrix removes any weighting effect that high magnitude data may have on the PCA model. The normalised matrix is given by:

$$X = \frac{X^0 - \bar{X}}{x_\sigma}$$

(11)

where $X^0$ is the original data matrix, $\bar{X} = 1_n \times [\bar{x}_1, ..., \bar{x}_n]$ and $x_\sigma = [x_{1\sigma}, ..., x_{n\sigma}]$ are the mean value vector and standard deviation vector of each column of $X^0$, respectively, and $1_n \in \mathbb{R}^n$ is a vector on ones, given that:

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$$

(12)

$$x_{j\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2}$$

(13)

The RPCA algorithm formulated by (Li et al., 2000), defines the dynamic characteristics of any process as the change in the mean, variance and correlation structure of the process variables with time. Hence, the objective of RPCA is to introduce time-variant updating mechanisms into the conventional PCA model structure in a sample-wise or batch-wise manner, depending on the application required. In addition to using mean and variance update of the PCA correlation matrix, adaptive determination of the principal components and monitoring indices ($T^2$ and $Q$ statistics) is necessary for real-time, on-line applications. A full explanation and derivation of the RPCA algorithm and the monitoring indices is provided by (Wade, 2004).

3 LabView Implementation

The implementation of the process monitoring and knowledge discovery tools for real-time application was performed within a Real-Time Control Platform (RTCP), (Sánchez et al., 2003), developed using the LabView software package, with a Data-logging and Supervisory Control (DSC) module allowing interfacing with an OPC (Object linking and embedding for Process Control) server. 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 Fig. 2.
The three units in the control platform are the Process Control Unit that monitors and tunes the control loops, the Human-Machine Interface (HMI) Unit, which is an interactive tool for visualisation of information collected in the system and the Data Quality Assessment (DQA) Unit, which performs the process monitoring and knowledge extraction tasks. Here, the development of the DQA for real-time implementation is shown with specific attention to the interactions between the several tools forming the DQA Unit.

### 3.1 Building the DQA Unit

LabView contains numerous functions and tools to aid the design of control or measuring devices for simulation or on-line engineering applications. These devices are referred to as Virtual Instruments (VIs) and are self-contained but highly versatile. The choice of LabView for the Real-Time Control Platform depended greatly on its ability to perform a large variety of tasks and the ease of communication between units and external devices.

For the DQA Unit, the tasks to be performed by the software are as follows:

- **Collect raw data from the plant**: Sensor signals are generated and transmitted to the PLC where they are digitised and stored in a registry. The digitised information is passed to the DQA via a tag index, which can be called in the VI to pass the data for processing in the DQA Unit;

- **Data pre-processing**: The initial training model is built off-line using real plant data collected from historical archives. The plant operator or process expert assists in assessing the data to ensure the training model is representative of the plant under normal, consistent and fault free conditions. Further pre-processing tasks may be performed on-line and can be supervised (i.e. measurement selection) or unsupervised (i.e. noise removal);

- **Data analysis**: The main process monitoring & fault detection, knowledge extraction and controller performance assessment algorithms are implemented within the LabView environment and may be executed in a sample-wise or batch-wise manner. A number of tools are available for the visualisation and reporting of the algorithm outputs;

- **Real-time implementation**: The data analysis tools must be able to adapt to the dynamic nature of the process. A Recursive Principal Component Analysis, (Li et al., 2000), algorithm was specifically designed to handle the transient data, for example. However, some
modification of the recursive algorithm code is required for on-line application within the LabView environment. A description of the algorithm used in this work is provided by Wade and Katebi, (Wade & Katebi, 2004).

- **Output information to other units**: The transference of information within the DQA Unit and to other units, such as the SCADA and HMI, is easily managed within LabView. Data-logging, reporting and error message functions can provide a historical overview of the system and DQA Unit performance over a specified period of time.

### 3.1.1 Data Input

The data measurements are passed to the RTCP using the DSC module via the OPC server. The Tag Engine enables each measurement that is passed to the platform to be read by the RTCP. Each measurement is stored in the PLC registry and assigned a tag index in the OPC, which may then be called by the DQA Unit as an input to the DQA VI, as shown in Fig. 3.

Once the data has been passed to the DQA Unit, the monitoring phase can begin. Initialisation of the process is achieved by passing initial state values to the unit prior to the collection of the measurements.

### 3.1.2 Data Pre-processing

A training model is required as an initial reference model for the process monitoring and knowledge extraction tools. The training model is built off-line in three stages (*data search & collection, data validation* and *training model development*) as shown in Fig. 4. By accessing historical data and trends, the plant operator or process expert can view information about the state of the plant over a desired period of time. Prior to building the training model a search and review of the historical data is performed with the aim of discovering a period of normal plant operation capturing the major trends in the data.

The problems with the supervised approach to data collection and validation are manifold. Firstly, the procedure is time consuming and requires an in depth knowledge of the process and plant, which relies on the aptitude of the operator or process expert. Secondly, the definition of the normal plant operating condition is partly subjective and may be a completely unrealistic term, given the vast criteria available for making such an observation. Finally, the size of the
initial model is restricted only by the duration of available records and, thus, an assumption that a certain period of time is suitable for capturing the most common trends observed in the processes must be made.

However, these arguments may be challenged by analysing the methodology. The DQA Unit combines a number of tools that are designed to be adaptive and recursive, through selective learning routines. If the initial model is not exactly representative of the normal operating condition, then the misclassification rate of the subsequent model may be high. This misclassification rate may be minimised in the validation step of the pre-processing stage, given the assumption that the operator or process expert base the data collection on some limited principle criteria, such as peak inflow, effluent quality or number of discharge consent failures. The selective learning process will result in the identification of a normal operating region with a high degree of likelihood and low misclassification rate.

3.1.3 Data Analysis and Real-time Implementation

The primary DQA Unit functions of process monitoring and fault detection (data-driven), knowledge extraction (knowledge-based) and controller performance assessment (model-based) are collated in a series of VIs and sub-VIs. The general structure of the unit is shown in Fig.5, highlighting the flow of information into the LabView environment, between the VIs within the unit and the subsequent output to other units and functions.

The data analysis functions are developed as a combination of LabView tools and some externally programmed code, typically Matlab. The core structure of the algorithms is generated and tested off-line, whereas on-line implementation requires the code be called and executed in the LabView environment, with peripheral utilities provided by the LabView tools. LabView has the ability to run Matlab code as a function in a VI using a Matlab Script Server, but expects a version of the Matlab software to be installed and running simultaneously with LabView. To overcome this problem, the generation of a Windows DLL (Dynamic Link Libraries) file using a suitable compiler enables a programme to be executed as stand-alone code.

The Matlab compiler is used to generate C/C++ code, which can be used to develop the DLL file. In LabView a Call Library Function Node is used to call the Matlab (or other) functions to the VI workspace. Input parameters such as the measurement data and statistical limits are passed to the node and output parameters are the desired outputs from the DLL function.
3.1.4 DQA Output

The outputs from the DQA are the indicators, charts and other visual tools that enable the operator to gain knowledge of the current status of the plant operation. A secondary set of outputs are the data that may be used by a Knowledge Discovery Unit (KDU), (Duzinkiewicz, 2002; Nielsen, 2002), and control system for knowledge discovery and control strategy scheduling, respectively.

Figure 6 shows the front-end user screen (Human-Machine Interface) by which an operator may access the DQA, (Sánchez et al., 2003). Here, real-time measurements and alarms may be monitored and recorded. The diagram shows the current configuration of the secondary treatment at Swinstie WWTP, with two anoxic tanks located to the left and the main aerated channels shown on the right. The Return Activated Sludge (RAS) and Waste Activated Sludge (WAS) pumps and oxygen blowers are located at the base of the screen. At the time of testing, the configuration of the plant was such that one anoxic tank and two aeration lanes were in operation. This was to compensate for the oversizing of the units when compared with the actual population equivalent.

As can be seen on the HMI, the current status of the pumps and blowers is indicated by the colour of the actuator, where red indicates in operation and green means the actuator is not in operation.

To access the DQA from the HMI, the Monitoring button, located on the Specialised Functions panel, is pressed. The user is then taken to the DQA front end as shown in Fig. 7. Here, the user can view the current status of the $Q$ and $T^2$ monitoring statistics, as well as a chart of the measurements used for monitoring, in real-time. A number of other functions are available on the interface. The training model may be extracted from a previously generated file and the size of the model can be selected using an input called Blocksize. A training model may be generated from current data by using the Force Update function, which will add new data to the model until the function is stopped, although this may then include erroneous data in the monitoring process.

Another function available is the ability to select the correlation matrix decomposition method. The user may select either SVD or Lanczos, which may be useful for comparing these different techniques over time. More information regarding these methods may be found in (Golub & Loan, 1983), for example.

The capacity to select which signals to use for monitoring and which to view is provided. This allows the user to switch the emphasis between signal locations, dependent on whether the
a priori reasoning, (Savic et al., 1999), for example.

The formation of an effective and hierarchical system structure at Swinstie WWTP is dependent on the quality and validity of the data that is collated at the SuCL.

The present limited control system implemented at Swinstie WWTP restricts the applicability of
A fast layer control and monitoring, for example, is focused on the performance of the Dissolved Oxygen (DO) control loop and monitoring of the secondary treatment phase. Ancillary monitoring can be extended to cover all measurements at the plant, with the possibility that the control system will be upgraded in the future to cover RAS, chemical dosing and nutrient control.

The KDU aims to resolve the complex and massive task of plant supervision with that of the operator’s role as a decision maker. The basic function of the KDU is to design a set of rules for decision-making by which the operator can assess the validity of the proposed action.

The schematic in Fig. 9 illustrates the function of the KDU within the supervisory control layer (SuCL). Data is stored in the SCADA database and pre-processed to remove outliers and missing values. Process monitoring and control functions are grouped together to indicate their continuous and collaborative application in the SuCL. The pre-processed data is used for sensor fault detection and identification (FDI) and process monitoring of the plant operation. These operations are performed on-line with RPCA, which has dual functionality, as described in (Wade & Katebi, 2004).

Information from the FDI is used to reconstruct faulty data and to inform the plant operator as to which sensor is causing the fault. The information from the process monitoring phase produces alarms, statistical charts and event logs, which are mostly unsupervised. However, the plant operator may be required to periodically monitor the performance of the plant and remedy the cause of alarms.

The KDU accepts information from the RPCA model and uses inductive learning and conceptual clustering techniques to provide plant operator support via rules for control decision-making. At present, Swinstie WWTP has minimal automatic control implemented and, thus, the role of the KDU is limited to a dialogue with the plant operator, in which information can be provided to initiate manual process control actions. The inductive learning process builds up a database of information that is utilised by the KDU. The initial development of the database uses information recovered from historical records and the measurement database and event log. The operator and process experts select mutually exclusive data sets from the available information, which relate to specific events that can be used to build a composite knowledge model of the plant operational states. When new information is received by the KDU, this knowledge model can be regularly updated to refine the operational state space, within which the rules are extracted.

Figure 10 shows how the KDU fits into the process monitoring and control system for Swinstie
WWTP. A development from Fig. 9, the schematic includes the control layers and performance and situation assessment module. The SuCL coordinates the transmission of control actions through the control layers, whereas the KDU handles the information from the monitoring, FDI and performance assessment applications. Because the KDU is part of the SuCL, information is shared mutually between them. For example, if there is a risk of not meeting a control objective, information processed through the KDU will communicate with the SuCL, advising it that a shift in the operation has occurred that requires a change in the control strategy. This may be simultaneously verified by alarms triggered in the monitoring phase. Predictive functions, such as Model Predictive Control (MPC), embedded at each control layer, and also provided by the KDU, can aid the assessment of risk over the prediction time horizon to reduce the number of false alarms or eliminate the delay between detection and action. The MPC developed for use at Swinstie WWTP is described fully in (Sanchez, 2004; Sanchez et al., 2004).

The recovery of knowledge and flow of information from the treatment plant to the monitoring and supervision levels is presented by the graphic, Fig. 11. The flowchart represents all the stages required for FDI, process monitoring and automatic rule-based supervision and control. The SCADA interface acts as the partition between information display and the latent tasks, such as data logging, alarm management and scheduling of monitoring operations. Information is passed to the SCADA in the form of reports, where it is logged and processed.

5 Real-time Monitoring

The RPCA algorithm was tested on-line at Swinstie WWTP over a three day period. Although a longer testing phase would have been desirable, it was limited by the time and measurement constraints.

5.1 Training model

A training model was developed by extracting data representing normal plant operation over a period of two weeks from the historical database located at Swinstie WWTP. Initially, the model contained all the available measurements from the secondary treatment phase of the process, including $DO$, ammonium ($NH_4$), phosphorus ($PO_4$), inflow and Mixed Liquor Suspended Solids $MLSS$. However, due to the problems detailed in the next section, only the four $DO$ measure-
ments were used for the testing phase, as shown in Fig. 12. The training model was generated over two weeks in order to capture as much information as possible about the diurnal and weekday-weekend trends, as can be seen from the figure.

5.2 On-line configuration

On-line analysis of the process monitoring algorithms was limited due to the following constraints:

- It was discovered that, at Swinstie WWTP, the blowers were oversized for the current operational DO demands. The resulting effect was that, even with one small blower running at its minimum speed (if the speed was lower there would be an increased risk of the motor stalling), the DO supply was higher than necessary. Turning off the blower would result in settling of suspended solids in the aeration tank, which would lead to clogging of the air diffusers at the base of the aeration lanes. Hence, the DO measurements were higher than the period during which the training model was developed;

- Subsequent analysis of the nutrient sensor measurements indicated that they were not reliable and required calibration;

- One of the two MLSS measurements in the aeration tank was faulty and the other was unreliable.

Hence, only the four Dissolved Oxygen measurements were selected for testing the monitoring algorithms. Figure 14 clearly shows the difficulties encountered when applying on-line testing. The DO measurements are mismatched with those from the training model in Fig. 12, with the mean value of DO3 and DO7 being approximately 3 mg/l greater in the test data than the training data and DO5 being almost anti-correlated with the other measurements.

5.3 On-line results

The algorithms for process monitoring, including fault detection, fault diagnosis and knowledge extraction, were implemented in the DQA unit of the real-time platform, described in section 3. However, due to the limitations detailed in section 5.2, only the fault detection and isolation (identification) capabilities could be tested on-line during the period available.
Figure 13 shows the flow and Dissolved Oxygen conditions during the three-day monitoring phase. During the monitoring period a number of rain events (0.2, 0.5, 0.7-1.3 and 1.7 days) occurred, which had the effect of reducing the DO measurements. The subsequent peaks in the DO values (shown clearly in Fig. 14) are indicative of the poor control and oversizing of the blowers, where too much oxygen is transferred to the aeration tank given the associated demand (a set-point of 1.8-2.2 mg/l based on the mean of the DO values) and, in some cases, low carbonaceous load in the influent due to dilution from the rainwater, i.e. the storm was not a first flush event.

To compensate for the initial mismatch between the training model and the real-time data, the model updating mode was initiated for a period of about 150 samples, until the monitoring statistics had decreased below their respective limits. As stated previously, in practice, this should not be done if the training data corresponds closely to the subsequent real-time measurements. In effect, by switching on the updating mechanism, the future measurements are used to re-build a training block that better represents the current operating condition. This action was only taken to enable testing of the algorithm, although this meant that the anti-correlated sensor (DO5) was subsequently trained as non-faulty, when in reality it is obviously generating spurious data.

The results from the on-line monitoring of the Dissolved Oxygen measurements are presented in Fig. 15. Initially, the monitoring statistics (upper graphs) show the model mismatch between the training data and the new input. This does not necessarily mean that there was a process upset or fault, but, instead, reveals that there was a substantial change in operating conditions at the plant since the training data was procured.

The results show that there were no subsequent instances of Q residual failure, but there were a number of periods where the $T^2$ statistic breached its 99th percentile limit. The Q residual is more sensitive to actual (sensor) faults and, as there were no occurrences of these during the monitoring phase, then the result was expected. However, the $T^2$ statistic is more representative of process upsets or changes in the process operating conditions and, here, it detected the change in DO caused by the ensuant reduction in load after the periods of high influent flowrate caused by heavy rainfall. Although not strictly a process upset, the results indicate the capacity for distinguishing between several operational scenarios, as discussed in Sanchez et al. (2004).

The contribution and scores plots (lower graphs) provide further information about the operating conditions during the monitoring phase. The contribution plot for each variable is plotted across the entire range of the sampling period and indicates which variable is most likely to be
impacting on the current operation. For example, it is most important to look at the periods when there is a disturbance or a fault. In this case, it is clear that Dissolved Oxygen measurement 5 (DO5) contributes mostly to the change in the DO state after the second and third storm periods. DO1 also contributes under these conditions and again during the first and last limit failures. The problem with interpreting the contributions result is that it is difficult to gain any knowledge about the fault or process upset with so few measurements. Although the contribution plot shows that DO5 is likely to be faulty, in reality a process upset would be manifest in the measurement linked with the abnormal condition, i.e. a flow sensor. Because this is not available, the contribution measure detects the anomaly between the faulty sensor and the other three fault-free sensors and this is expressed in the plot.

The scores plot is useful for visualising the distance and magnitude of the scores from the nominal operating point (0, 0). In real-time, it is possible to see if the scores are moving towards or away from the nominal point and in which direction. This is useful for diagnosing particular events, such as faults or process disturbances. Clustering and classification tools can be used to define regions in the scores plot associated with these particular phenomenon and the mathematical analysis of the direction and magnitude of the fault can be used to recover the best estimate for use by the controller algorithms, over the period that a fault is detected. To emphasise this point, Fig. 16 shows the $T^2$ scores plot with the 95%ile and 99%ile limits shaded. It can be seen that the out-of-limit scores are located at three extrema, which are, in addition, situated in three separate quadrants of the scores plot, with (0, 0) representing the centre of the quadrants. Further analysis shows that the initial out-of-limit score (sample 1), representing the training model mismatch, has a position in the lower left quadrant, whilst the largest $T^2$ value (sample 1060), representing the second peak DO, is located in the top left quadrant. Subsequently, it can be seen that different scenarios may be classified according to the location of their scores within the plot. It should be noted that in part of the plot some of the 95%ile limit failures are closer to the nominal point than the non-failures. This is because the RPCA algorithm has adaptive limits, which can be seen by the $T^2$ plot in Fig. 15. The region represents the change in limit after sample 400, where the decomposition technique was switched from SVD to Lanczos in the DQA.

Another interpretation of the results may be achieved by looking at a biplot of the $T^2$ statistic versus the $Q$ statistic, as shown in Fig. 17. Here, the $T^2$ confidence limits are indicated by green (95%ile) and red (99%ile). Again, the model mismatch is indicated as sample 1 and the $DO$
increase after the extreme rain event by 1060. It is interesting to note that events can be traced by intuitive appreciation of the direction of the biplot values, where the initial model updating can be tracked as moving from sample 1 towards the origin, whereas a change in the process is shown by a vertical shift in the values of the $T^2$ statistic.

6 Conclusions

A LabView environment was created to provide real-time control and monitoring capacity for testing the RPCA algorithms on-line at a wastewater treatment plant. The DQA module provided visualisation of the monitoring statistics as well as other functions for specific tasks, such as model training, data logging and fault identification. Although a full implementation of the Supervisory Control Layer was not feasible, a discussion of its capabilities and advantages was provided, with a view to utilisation in the context of fault monitoring and knowledge extraction.

The RPCA algorithm was tested on-line for three days using a training model generated from two weeks of data. From initial observations, it was clear that the current plant operation at the time of testing had changed considerably since the training model data was generated. However, this was not a problem because i) it highlighted the fact the RPCA algorithm had correctly detected a problem and ii) a function within the DQA was used to re-train the model on-line.

The results from the testing phase confirmed that the $T^2$ statistic was more sensitive to process changes than the $Q$ statistic, although the limited available measurements did cause problems in interpretation of the results. Further analysis also showed the limitations of the contribution plot for fault isolation, although other diagnosis tool are developed and available for real-time testing given appropriate data availability. The characteristics of the scores plot were also utilised and it was shown that distinct process events could be discriminated by location in the scores space.

It must be noted that the limitations caused by data availability and the infeasibility of some measurements has restricted the full testing of the process monitoring scheme. However, the combination of off-line and on-line testing has highlighted the advantages of an all-encompassing recursive tool, such as RPCA. The benefits of using combination of tools, such as discriminant analysis and inductive learning, is only relatively exploited in practice. The significant factor for the appropriate and robust use of these tools is the acquisition of reliable and representative data from the plant. This means an inter-disciplinary approach is required to obtain the maximum benefit from the preliminary training and pre-processing stages.
An interesting conclusion from this is, as the wastewater industry is moving towards fully automated plants, then the number of process operators or experts available to integrate with the control/software engineers will be increasingly restricted. Hence, knowledge availability of the data required for monitoring may also be restricted. One possible area for future research is the integration of process models with unsupervised learning to automatically generate valid data sets for pre-monitoring training model development.

**References**


Figure 2: Real-time Control Platform interface with the SCADA and PLC (adapted from Sánchez et al. (2003))
Figure 1: Decomposition of the original data matrix into the PCA model (process) space and residual (noise) space (adapted from Rosén (2001))

Figure 3: Data input path for the DQA Unit
Figure 4: Training model development using supervised data processing

Figure 5: The data analysis tools provided by the DQA Unit
Figure 6: Human-Machine Interface (HMI) of the RTCP
Figure 7: DQA front end screen-shot
Figure 8: Fault Identification/Diagnosis screen-shot
Figure 9: SCADA-WWTP schematic indicating the role of the KDU and monitoring tools
Figure 10: The operational planning schematic developed for the SMAC Project
Figure 11: Information flowchart for operational planning developed for the SMAC Project
Figure 12: Dissolved Oxygen training data for on-line monitoring
Figure 13: Flow and Dissolved Oxygen measurements collected during the testing phase
Figure 14: Dissolved Oxygen measurements during the monitoring phase
Figure 15: Fault detection and contribution and scores plots for 3 days on-line, real-time monitoring of a WWTP
Figure 16: Scores plot for on-line monitoring showing alarms
Figure 17: Biplot of the $T^2$ vs. $Q$ statistics showing alarms