VERSITA

Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 4, 939–949
DOI: 10.2478/v10006-012-0070-1

amcs

DATA–DRIVEN MODELS FOR FAULT DETECTION USING KERNEL PCA: A WATER DISTRIBUTION SYSTEM CASE STUDY

ADAM NOWICKI, MICHAŁ GROCHOWSKI, KAZIMIERZ DUZINKIEWICZ

Faculty of Electrical and Control Engineering
Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
e-mail: adam.j.nowicki@gmail.com,{m.grochowski,k.duzinkiewicz}@ely.pg.gda.pl

Kernel Principal Component Analysis (KPCA), an example of machine learning, can be considered a non-linear extension of the PCA method. While various applications of KPCA are known, this paper explores the possibility to use it for building a data-driven model of a non-linear system—the water distribution system of the Chojnice town (Poland). This model is utilised for fault detection with the emphasis on water leakage detection. A systematic description of the system's framework is followed by evaluation of its performance. Simulations prove that the presented approach is both flexible and efficient.

Keywords: machine learning, kernel PCA, fault detection, monitoring, water leakage detection.

1. Introduction

This paper discusses the concept and results of applying kernel principal component analysis to the problem of leakage detection in Water Distribution Systems (WDSs). The motivation behind it is twofold. Firstly, leakages in water distribution systems are a serious problem all over the world. The level of losses in urban water networks can be measured as the difference between the volume of water delivered to this system and the volume of billed authorised consumption. This difference, referred to as non-revenue water, reaches 48 billion m^3/year (Thornton *et al.*, 2008) according to the World Bank. The problem does not only concern developing countries. European Commission studies indicate that in certain areas of Europe losses can be as high as 50%. A major source of losses are leakages. Since water distribution systems are located underground, inspection is troublesome. Leaking fittings or pipe-cracks are difficult to spot and, as a consequence, leakages may remain unnoticed for years.

Secondly, kernel methods have recently become popular in applications typical for machine learning such as computer vision (Hott, 2008; Li *et al.*, 2008), voice recognition (Campbell *et al.*, 2006), clustering (Kulczycki and Charytanowicz, 2010) or medicine (Lima and Coelho, 2011). However, the number of studies looking into the possibility of applying kernel methods to fault detection in industrial man-made systems remains limited (Slišković *et al.*, 2011). Fault detection systems can serve a great help for Fault Tolerant Control (FTC). They can isolate a particular fault or just indicate in which operational state the system is. The FTC mechanism can then properly modify the controller (e.g., by adapting the performance index and the constraints in the MPC approach) or replace it with another one, better fit to the actual plant operating condition. The examples of such approach can be found, e.g., in the works of Patan and Korbicz (2012) or Brdyś *et al.* (2008).

The approach presented in this article is based on data-driven novelty detection, where the monitoring system learns during the training phase the correct behaviour of the system; in other words, it creates a black-box model of a correctly operating system. Later on, during the generalisation phase, the corresponding operational states are regarded as normal states while novel, previously unseen operational states are regarded as faults. This seems an interesting alternative to the typical modelling based on physical laws because no in-depth knowledge of the system considered is required. It has also some advantages over the classification approach (popular in machine learning), where symptoms of a fault are sought—it is possible to detect unforeseen faults and training data do not need to be rich in samples representing faulty states.

Application of a data-driven novelty detection system to the problem of leakage detection has already

been studied (Duzinkiewicz *et al.*, 2008; Jezior *et al.*, 2007). However, a major limitation came from the linear nature of the method applied—Principal Component Analysis (PCA). As a result, the system led to limited sensitivity or a high false alarms rate. Although an attempt to utilise a non-linear extension of PCA—Kernel Principal Component Analysis (KPCA)—is found in the work of Nowicki and Grochowski (2011), this paper presents a systematic and comprehensive approach as well as a quantitative performance comparison between PCA and KPCA.

The remainder of this paper is organised as follows. Section 2 presents a state-of-the-art review of the KPCA algorithm for data-driven novelty detection. It starts with a general overview of the KPCA method. Next, four aspects of model building and usage are covered: training data processing, an algorithm for data driven modelling, test data processing, and an algorithm for determining if the test data are represented by the model. Section 3 presents the framework of the proposed fault detection system. The first subsection presents the approach to leakage detection; the second one explains how the algorithms presented in the previous section are applied to the problem of leakage detection in water distribution systems. Some of the ideas (node selection, fault symptom propagation) described in this section are based on the previous work, while other have been revised to take into account the non-linear KPCA method. The section ends with a detailed presentation of the steps taken to build the proposed system. Section 4 presents results of experiments and, finally, Section 5 summarises the paper.

2. Kernel PCA for fault detection

2.1. General premise. This section presents state-of-the-art kernel PCA based on the novelty detection approach. The algorithms and notation presented here are crucial to comprehensively understand the operating principle of the proposed system. This section has been founded on the papers by Schölkopf *et al.* (1998), Shawe-Taylor and Cristianini (2004), as well as Hoffman (2007).

Principal component analysis allows finding a linear subspace of the input space where a data set can be represented with a minimal loss of variance. This allows detecting linear patterns (Jackson, 1991). Kernel PCA, described by Schölkopf *et al.* (1998), can be considered a non-linear extension of PCA that combines multivariate analysis and machine learning. Instead of looking for a linear relationship between the variables in the input space $\mathbb{R}^n$, all measurements are mapped into a higher dimension inner product space $\mathbb{F}$, called the feature space, through a non-linear mapping ϕ where linear patterns are sought using PCA

$$\phi : \mathbb{R}^n \to \mathbb{F}. \tag{1}$$

The linear relationship in the feature space corresponds to the nonlinear relationship in the input space. However, thanks to the kernel trick introduced by Aizerman *et al.* (1964), neither mapping ϕ nor any image $\phi(x)$ needs to be calculated explicitly. This allows not only significant reduction of computational workload, but one does not have to look for ϕ that linearises the problem, either. The trick can be applied to any algorithm that operates on data entirely in terms of the inner product $\langle \cdot, \cdot \rangle$. Then it is possible to substitute it with a kernel function $\kappa(\cdot, \cdot)$. This corresponds to inner product $\langle x, z \rangle$ in the input space $\mathbb{R}^n$ replaced by $\langle \phi(x), \phi(z) \rangle$ in the feature space $\mathbb{F}$ which is given by $\kappa(x, z)$:

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle. \tag{2}$$

The kernel function κ has to fulfil Mercer's theorem to ensure that the mapping ϕ exists (Mercer, 1909).

The choice of the function can be dictated by a domain-specific knowledge. The Gauss function is used in this paper:

$$\kappa(x, z) = \exp\Big(\frac{-\|x - z\|^2}{2\sigma^2}\Big). \tag{3}$$

This distance-based kernel allows generalising the data without any prior assumptions regarding relationships between variables. The resulting feature space $\mathbb{F}$ associated with this kernel function is infinite-dimensional. However, in order not to over-complicate the remainder of the paper, the empirical feature space $\mathbb{R}^m$ associated with empirical mapping ϕ_m is used instead (Schölkopf *et al.*, 1999; Xiong *et al.*, 2005):

$$\phi_m : \mathbb{R}^n \to \mathbb{R}^m. \tag{4}$$

Like ϕ, the mapping ϕ_m does not need to be known explicitly. For a data set consisting of m data points, $\mathbb{R}^m$ represents an m-dimensional Euclidean space that preserves the geometrical structure in $\mathbb{F}$ with inner product defined as

$$\kappa(x, z) = \langle \phi_m(x), \phi_m(z) \rangle. \tag{5}$$

This means that PCA carried out in the feature space $\mathbb{F}$ and in the empirical feature space $\mathbb{R}^m$ gives the same result (Schölkopf *et al.*, 1999).

2.2. Preparing training data. Let x_i be a $1 \times n$ row vector containing measurements that represent the operational state of the system considered at a given time:

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{in} \end{bmatrix}. \tag{6}$$

This vector belongs to the input space $\mathbb{R}^n$. A training set, consisting of m data points, is represented by an $m \times n$

matrix X:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_{11} & & x_{1j} & & x_{1n} \\ & \ddots & & & \\ x_{i1} & & x_{ij} & & \\ & & & \ddots & \\ x_{m1} & & & & x_{mn} \end{bmatrix}. \tag{7}$$

It is assumed that the training set consists of vectors representing certain behaviour of the modelled node and that all measured variables are normalised, i.e., column vectors of X have unit variance, thus fulfilling

$$\frac{1}{m-1}\sum_{i=1}^{m}(x_{ij}-\mu_j)^2 = 1 \quad \text{for } j = 1, 2, \ldots, n, \tag{8}$$

where

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}.$$

In order to use the kernel trick, an algorithm must operate on data entirely in terms of the inner product. In the case of PCA, there is a dual formulation that fulfils this requirement; the algorithm is fed with Gram's matrix G where each element is calculated as an inner product between data points $g_{ij} = \langle x_i, x_j \rangle$. In KPCA, this matrix is substituted with the kernel matrix K:

$$K = \begin{bmatrix} k_{11} & & k_{1j} & & k_{1m} \\ & \ddots & & & \\ k_{i1} & & k_{ij} & & \\ & & & \ddots & \\ k_{m1} & & & & k_{mm} \end{bmatrix}, \tag{9}$$

where $k_{ij} = \kappa(x_i, x_j)$.

Typically, PCA operates on centred data (Jackson, 1991) and therefore the image of the training set $\phi_m(X)$ in the empirical feature space $\mathbb{R}^m$ needs to be centred. Although the image is not known explicitly, the centreing procedure can be applied directly to the Kernel matrix.

Let ϕ_S denote the centre of the mass of the training set X in the empirical feature space $\mathbb{R}^m$:

$$\phi_S = \frac{1}{m}\sum_{i=1}^{m}\phi_m(x_i), \tag{10}$$

and let $\widetilde{\phi}_m$ be the empirical mapping that takes into account the centreing procedure:

$$\widetilde{\phi}_m(x) = \phi_m(x) - \phi_S. \tag{11}$$

For any two data points x and z, it is possible to evaluate the inner product between centred images (Schölkopf *et al.*, 1998; Shawe-Taylor and Cristianini, 2004):

$$\begin{aligned} &\langle \widetilde{\phi}_m(x), \widetilde{\phi}_m(z) \rangle \\ &\quad = \langle \phi_m(x) - \phi_S, \phi_m(z) - \phi_S \rangle \\ &\quad = \langle \phi_m(x), \phi_m(z) \rangle - \langle \phi_m(x), \phi_S \rangle \\ &\qquad - \langle \phi_S, \phi_m(z) \rangle + \langle \phi_S, \phi_S \rangle \\ &\quad = \kappa(x,z) - \frac{1}{m}\sum_{i=1}^{m}\kappa(x,x_i) - \frac{1}{m}\sum_{i=1}^{m}\kappa(z,x_i) \\ &\qquad + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\kappa(x_i,x_j). \end{aligned} \tag{12}$$

Therefore, the centreing procedure can be applied directly to a kernel matrix. In a matrix-wise notation, this corresponds to

$$\begin{aligned} \widetilde{K} = K - 1\left(\frac{1}{m}1^T K\right) - \left(\frac{1}{m}K1\right)1^T \\ + 1\left(\frac{1}{m^2}1^T K 1\right)1^T \end{aligned} \tag{13}$$

where 1 and $\widetilde{K}$ denote the $m \times 1$ all-ones vector and a kernel matrix K centred at the origin of the coordinates in $\mathbb{R}^m$, respectively. This normalisation can be associated with a mapping $\widetilde{\phi}_m$ that takes into account this centreing. Having represented the training data in the form of a kernel matrix $\widetilde{K}$, it is possible to process it using PCA.

2.3. Looking for a pattern: The data model. By applying PCA to a training set, one can find an orthonormal coordinate system where subsequent coordinates (for historical reasons they are often called scores), evaluated for data points from this set, have decreasing variation. The primal formulation of PCA (Jackson, 1991) allows calculating these scores using eigenvectors V and eigenvalues Λ of the covariance matrix C calculated for the training data X in the empirical feature space $\mathbb{R}^m$:

$$C = \frac{1}{n-1}\widetilde{\phi}_m(X)^T\widetilde{\phi}_m(X). \tag{14}$$

Let λ_i be the i-th eigenvalue from the diagonal matrix Λ corresponding to v_i, the i-th eigenvector from V. Then, assuming that the eigenvalues and the eigenvectors are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, the column vectors of V are a new basis that spans the PC space. Therefore, the image of the training set $\widetilde{\phi}_m(X)$ could be projected on the eigenvectors V by

$$X^{PC} = \widetilde{\phi}_m(X)V \tag{15}$$

However, in the case of the kernel PCA the covariance matrix C cannot be calculated explicitly because $\widetilde{\phi}_m(X)$ is unknown, and therefore the vectors of

V cannot be evaluated directly. Instead, it is possible to use the dual formulation of PCA (Barshan *et al.*, 2011; Shawe-Taylor and Cristianini, 2004). The eigenvectors V and the eigenvalues Λ of the re-scaled covariance matrix $\widetilde{\phi}_m(X)^T\widetilde{\phi}_m(X)$ can be obtained from the kernel matrix $\widetilde{K} = \widetilde{\phi}_m(X)\widetilde{\phi}_m(X)^T$, thus allowing to use the data in terms of the inner product. Using Singular Value Decomposition (SVD), the image of the training set $\widetilde{\phi}_m(X)$ can be decomposed as

$$\widetilde{\phi}_m(X) = U\Lambda^{\frac{1}{2}}V^T, \tag{16}$$

where U and V contain left and right singular vectors given as column vectors that are eigenvectors of $\widetilde{\phi}_m(X)^T\widetilde{\phi}_m(X)$ and $\widetilde{\phi}_m(X)\widetilde{\phi}_m(X)^T$, respectively, while $\Lambda^{-\frac{1}{2}}$ holds the square roots of eigenvectors $\widetilde{\phi}_m(X)^T\widetilde{\phi}_m(X)$ which are identical to the eigenvectors of $\widetilde{\phi}_m(X)\widetilde{\phi}_m(X)^T$. This allows representing V as

$$V = \widetilde{\phi}_m(X)^T U\Lambda^{-\frac{1}{2}}. \tag{17}$$

By combining (15) with (17), it is possible to calculate the PC scores X^{PC} of the training set:

$$X^{PC} = \widetilde{\phi}_m(X)\widetilde{\phi}_m(X)^T U\Lambda^{-\frac{1}{2}} = \widetilde{K}U\Lambda^{-\frac{1}{2}}, \tag{18}$$

where U and Λ are the eigenvectors and the eigenvalues of $\widetilde{K}$.

If the linear pattern exists within the data in the feature space, then only the first r coordinates are sufficient to describe the data, and so only the first r eigenvalues and eigenvectors need to be extracted. Let U_r denote the matrix containing first r column vectors from U and Λ_r denote the matrix containing corresponding eigenvalues across its diagonal. The first r PC scores of the training set X can be calculated as

$$X^{PC(r)} = \widetilde{K}U_r\Lambda_r^{-\frac{1}{2}}. \tag{19}$$

The aim is to find a configuration of relevant eigenvectors r and parameter σ of the kernel function κ that allows describing the general pattern represented by the training set X exclusively. This means that the transformation would be well fitted for points that follow the discovered pattern. Thus, little information is lost while projecting on V_r. Great information losses take place in all other cases. Hence, the training set X, the kernel function κ and the matrices U_r and Λ_r define the model.

2.4. Preparing test data. Let X_{new} be a test set that needs to be matched against the pattern represented by the kernel PCA model. It is given as a $p \times n$ matrix containing p data points $x_{new\,i}$ given as $1 \times n$ row vectors:

$$X_{new} = \begin{bmatrix} x_{new\,1} & \cdots & x_{new\,i} & \cdots & x_{new\,p} \end{bmatrix}^T. \tag{20}$$

It is assumed that X_{new} is normalised with respect to the training set X. Like the training set, these data need to be represented in the form of a kernel matrix. Let this $p \times m$ matrix be denoted by K_{new}. It contains evaluation of a kernel function between each test point $x_{new\,i}$ and each training point x_j:

$$K_{new} = \begin{bmatrix} k_{n\,11} & & k_{n\,1j} & & k_{n\,1m} \\ & \ddots & & & \\ k_{n\,i1} & & k_{n\,ij} & & \\ & & & \ddots & \\ k_{n\,p1} & & & & k_{n\,pm} \end{bmatrix}, \tag{21}$$

where $k_{n\,ij} = \kappa(x_{new\,i}, x_j)$. K_{new} needs to be normalised (centred in the feature space) with regard to the training set by

$$\begin{aligned} \widetilde{K}_{new} = K_{new} &- 1_p\left(\frac{1}{m}1_m^T K\right) \\ &- \left(\frac{1}{m}K_{new}1_m\right)1_m^T \\ &+ 1_p\left(\frac{1}{m^2}1_m^T K 1_m\right)1_m^T, \end{aligned} \tag{22}$$

where 1_p denotes the $p \times 1$ all-ones vector and 1_m denotes the $m \times 1$ all-ones vector.

Additionally, evaluation of the kernel function between each test vector and itself is needed. This is stored in the $p \times 1$ vector K_{ext}:

$$K_{ext} = \begin{bmatrix} k_{e\,1} & \cdots & k_{e\,i} & \cdots & k_{e\,p} \end{bmatrix}^T, \tag{23}$$

where $k_{e\,i} = \kappa(x_{new\,i}, x_{new\,i})$. Following (12), this vector is subject to the centreing procedure:

$$\begin{aligned} \widetilde{K}_{ext} = K_{ext} &- \left(\frac{2}{m}K_{new}1_m\right) \\ &+ 1_p\left(\frac{1}{m^2}1_m^T K 1_m\right). \end{aligned} \tag{24}$$

2.5. Matching a new data point against the pattern: The reconstruction error. Following evaluation of reconstruction error as described by Hoffman (2007), it is possible to measure the squared distance $E(x_{new})$ between the projection x_{new}^{PC} of a test point x_{new} on all principal components V and its projection $x_{new}^{PC(r)}$ on relevant eigenvectors V_R using the Pythagoras theorem:

$$E(x_{new}) = \|e\|^2 = \|x_{new}^{PC}\|^2 - \|x_{new}^{PC(r)}\|^2. \tag{25}$$

A small value of the error E indicates that the test point is well represented by the model, which means that it follows the pattern.

The term $x_{new}^{PC(r)}$, an element of $X_{new}^{PC(r)}$, can be easily calculated using (19):

$$X_{new}^{PC(r)} = \widetilde{\phi}_m(X_{new})\widetilde{\phi}_m(X)^T U_r \Lambda_r^{-\frac{1}{2}} = \widetilde{K}_{new} U_r \Lambda_r^{-\frac{1}{2}}, \quad (26)$$

but evaluation of the first term x_{new}^{PC} using (18) could impose some serious workload. However, it is possible to use two facts here. Firstly, since PCA operates in the feature space on centred data, it can be perceived as a pure rotation of the basis without any translation. Secondly, it is not the vector itself that is required here, but the squared distance to the origin. This means that x_{new}^{PC} is equal to the inner product $\langle \widetilde{\phi}_m(x_{new}), \widetilde{\phi}_m(x_{new}) \rangle$, which is already calculated in (24) and denoted by $\widetilde{K}_{ext}$.

Using (26) and (24) it is possible to evaluate the reconstruction error E in (25). The iso-potential curves or surfaces of the reconstruction error can serve as decision boundaries for novelty detection (Hoffman, 2007).

3. Fault detection: The system framework

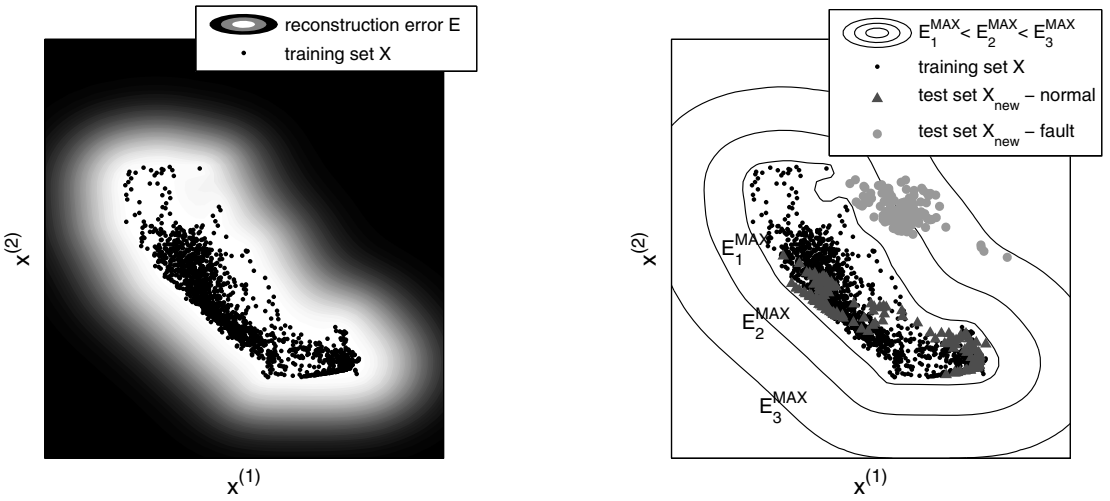
3.1. General overview of the framework. Although there are various approaches to fault detection (Venkatasubramanian *et al.*, 2003a; 2003b; 2003c), the data-driven approach has been chosen for purposes of this paper. Similar attempts have already been made with regard to water distribution systems (Mashford *et al.*, 2009; Duzinkiewicz *et al.*, 2008). The most popular approach for detecting leakages is the acoustic based method, which uses the fact that leakages produce sound of a certain frequency; an overview of other methods is given by Xiao-Li and Jiang Chao-Yuan (2008). These methods cannot be adopted in a straightforward manner for a fault detection system since they require manual measurements to be taken at various points across the network. The approach presented in this paper utilises measurements taken in some fixed nodes—pressure and flows associated with each node registered in a cyclic manner. The analysis of these physical quantities allows, to a certain extent, detecting faults that introduce perturbation into the network. Among these faults leakages are the most common and serious as well.

There are two main reasons for that data-driven approach. Firstly, the values of flows and pressure measured in any node of real-life networks are, in general, highly repeatable on a daily basis and they predictably vary depending on the season. During a leakage the relationship between measurements is disturbed. Thus the measurements themselves can directly provide a symptom of a fault.

Secondly, a water distribution system is a dynamic, complex, nonlinear system with varying parameters. The classical approach, based on control charts or modelling and estimation has been used for simple pipeline systems (Isermann, 1984). However, present-day water distribution systems require more powerful methods. Data-driven machine learning methods allow us to disregarding, to a certain extent, the complexity of the modelled network—the problem of describing the system itself is simplified to that of describing data that represent the behaviour of the system in question. This translates to several benefits. To begin with, thanks to the fast dynamics of the hydraulics, it is possible to consider measurements of flow and pressure taken at a given point in the network as the vector representing the operational steady state (transient states can be neglected) of the network at that point. Additionally, quick propagation of the perturbance caused by the fault allows to process separate measurement vectors independently and locate the fault quickly. Another benefit is the possibility to decompose a large system into a number of smaller systems (Jezior *et al.*, 2007), each modelled and monitored independently, as will be explained later. In order to deal with nonlinear relationships between variables in the training data (the result of non-linearity in the represented system), a method allowing one to capture non-linear patterns needs to be utilised. Finally, the network's parameters are subject to changes, resulting in different data representations of the system, but the process typically spans years, so the model does not require frequent updates.

A water system can be represented by a set of nodes (junctions, tanks or reservoirs) connected by links (pipes, pumps or valves). It takes the form of a network. As proven by a series of experiments (Jezior *et al.*, 2007), leakage symptoms (perturbation of the relation between pressure and flows) propagate over the network within a limited distance from the source of the leakage. There is no easy way to determine this distance but, generally speaking, the larger the leakage, the stronger the leakage symptom and, consequently, the distance grows. As the damping of the symptom varies across the network, the symptom propagates with a different magnitude in different directions. The mains in particular (the network's backbone) have been identified to have strong dumping properties. This means that, in order to be able to detect faults with reasonable efficiency, a number of nodes where the measurements are taken is required. Promising candidates are the nodes where several pipes are crossing and which supply water to a few other nodes. Unfortunately, there is no straightforward way to select nodes that would guarantee to perform well. It is also not possible to determine *a priori* how many measurement points are required to provide the assumed quality of detection. Hence the nodes have to be selected empirically (Jezior *et al.*, 2007).

The nodes where pressure and flows are measured are called monitoring nodes. For each monitoring



(a) (b)

Fig. 1. Graphical interpretation of reconstruction error E (bright area corresponds to low values of E) (a), three sample reconstruction error thresholds E^{MAX} (only E_1^{MAX} provides good leakage detection) (b). The training set represents two flows (variables $x^{(1)}$, $x^{(2)}$) at node 030.

node a separate model is built. This corresponds to splitting a complex network into a number of small semi-independent subsystems. Each monitoring node provides information about faults only with respect to its local neighbourhood.

At each time step, it is possible to observe n measurements ($n-1$ flows and a pressure) for a single monitoring node with $n-1$ links—a single set of n measurements can be perceived as a data point in an n-dimensional input space $\mathbb{R}^n$. A set of m data points collected over a period of time from a single node serves as a training set X. That set should be representative, which means that it should be rich enough to contain information about all key operational states of the monitoring node. The training set representing a given node is perceived as a set of points in $\mathbb{R}^n$ (Fig. 1(a)). It can be enclosed by a regular decision boundary that, on the one hand, tightly encloses this set, but on the other hand, generalises the relationship between the variables (Fig. 1(b)). Provided that the training set is representative, any data point lying outside the decision boundary would represent a faulty state. In this paper, kernel principal component analysis is employed to build the said decision boundary.

Let E^{MAX} be the threshold value of the reconstruction error chosen for a given KPCA model. Following the previous reasoning, for any test vector x_{new},

$$E(x_{new}) \begin{cases} \geq E^{\mathrm{MAX}} \Rightarrow & x_{new} \subset \text{faulty states}, \\ < E^{\mathrm{MAX}} \Rightarrow & x_{new} \subset \text{normal states}. \end{cases} \tag{27}$$

The graphical interpretation of (27) is the iso-potential surface in $\mathbb{R}^n$ enclosing the training set that serves as a decision boundary (Fig. 1).

3.2. Set-up of the network model. This section describes the steps necessary to set up the proposed fault detection system (Fig. 2). The entire process starts with selection of monitoring nodes. As described earlier, this is a purely heuristic operation (indicated by the trapezoidal shape of the operation in Fig. 2). It requires the knowledge of the network's structure and understanding of fluid mechanics, or a simple model of the network that enables to verify which nodes are sensitive to water distribution changes. It might be a good idea to identify semi-independent zones with nodes of similar characteristics first (e.g., a residential area within a single district). Those zones might be separated by the network's backbone or separated physically as district metering areas (Thornton *et al.*, 2008). Within each zone at least one monitoring node should be selected, but depending on the size of this zone, the required sensitivity of the system and performance of selected nodes, more monitoring nodes might be needed.

Once the monitoring nodes are identified, a separate model is built for each node. The remaining part of this subsection describes operations for setting up a model of a single node, to be repeated for each node.

It is required that, in the monitoring node, cyclic measurements of pressure and flows in all incoming pipes associated with this node be taken as described earlier. The training data must be acquired from a fault-free system to guarantee that the training set represents correctly the operating sub-system. The test set should

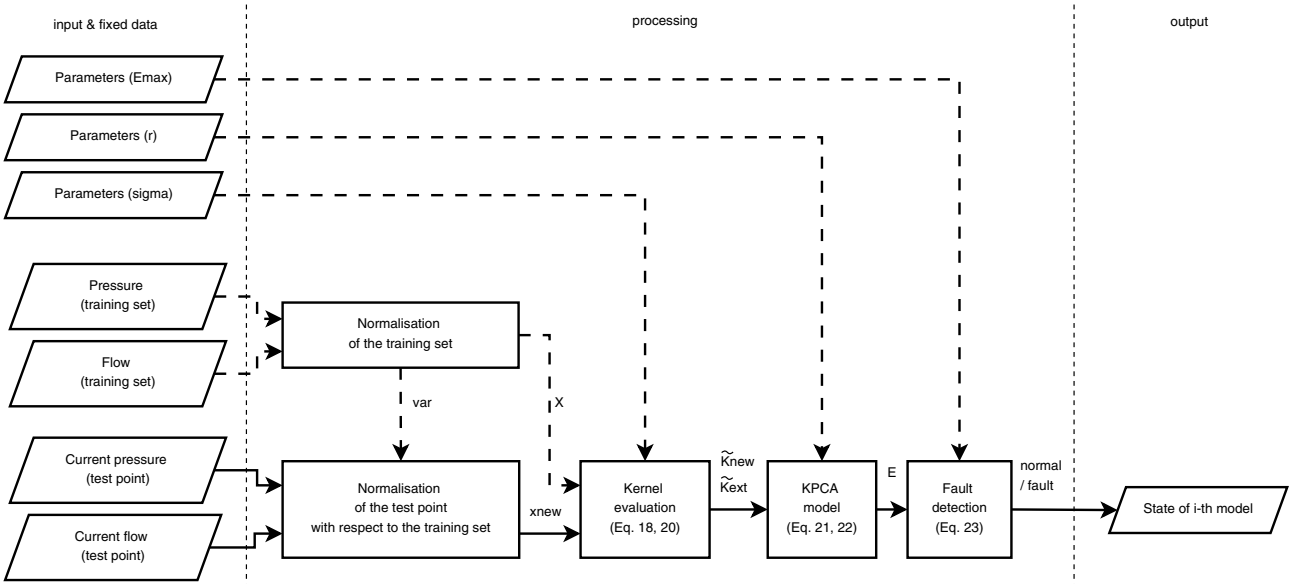


Fig. 3. Data processing model for fault detection. The solid line denotes variable input, the dashed line denotes fixed input.

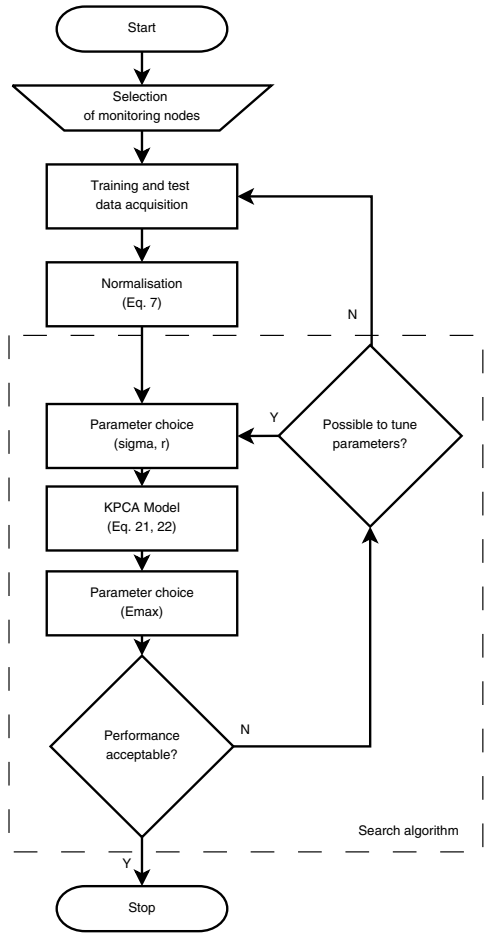


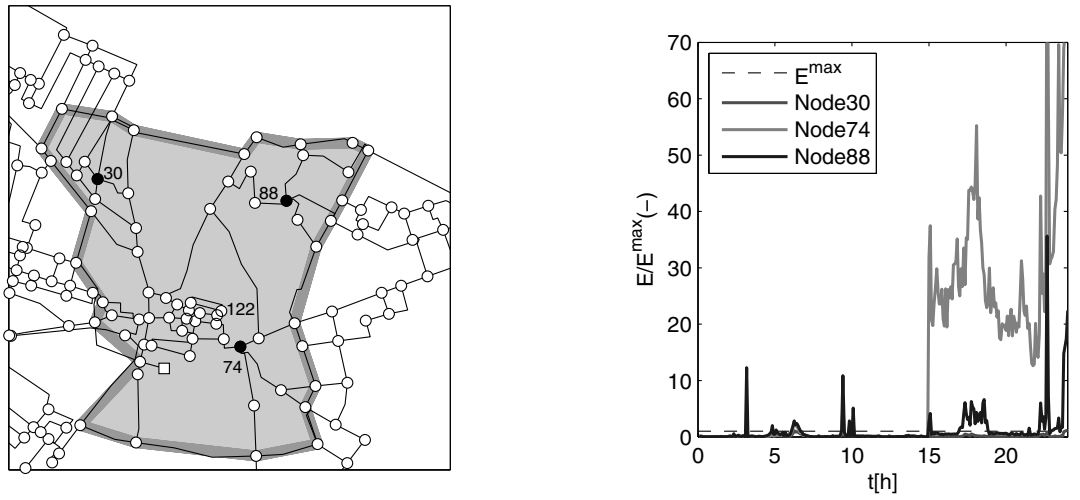
Fig. 2. Flowchart for setting up a WDS fault detection system.

represent the faults that are expected to be detected by the system. Both the training and test sets are then normalised with respect to the training set (as in Eqns. (8) and (22)), to ensure that variables are equal in the statistical sense.

In the next step, two parameters, namely, σ and r, have to be chosen. Large values of σ allow building a generalised model of the data that results in a fault detection capability similar to the one provided by the classic linear PCA model, while small values of σ and a high number of relevant eigenvectors r result in a rather strict data model. For given parameters it is possible to build a KPCA model and evaluate its behaviour for the training set. This can provide the first estimate of E^{MAX}. It should be, on one hand, large enough to minimise false alarms during normal operation, but on the other—small enough to be able to detect abnormalities during a fault. For a well-prepared KPCA model, E^{MAX} can be chosen to provide some fixed performance over the training set, e.g., as the 95th percentile of the reconstruction error calculated for the training set.

Finally, it is possible to evaluate the efficiency of the monitoring node model by verifying the reconstruction error calculated for the test data. If the performance is not satisfactory, one can try to look for different sets of σ and r parameters, then rebuild the KPCA model and go again through the E^{MAX} procedure. However, to avoid the manual trial and error approach, it is possible to use a search algorithm instead. The grid search method is particularly popular for kernel methods, but other strategies can be applied as well (Nogayama *et al.*, 2003).

If the performance turns out below expectations and it cannot be further improved by tuning the parameters, either the training set must be re-examined or the



(a) (b)

Fig. 4. Model of the Chojnice WDS with the residential zone (grey background) chosen for the performance evaluation and three monitoring nodes: 030, 074, 088 (black circles) (a), reconstruction error values evaluated for a sample test scenario (leakage of a size $1.4\,\mathrm{m}^3/\mathrm{h}$ simulated in node 122 between 15 and 24 h) for the three monitoring nodes (b).

expectations are too high. Should the latter be the case, the test set needs to be updated to represent the new, revised expectations.

Figure 3 illustrates the data processing model for fault detection. For each monitoring node an independent model is created. The model is defined by the training set X with some fixed variance var and a set of three parameters: r, σ and E^{MAX}. They are all fixed (this is indicated by the dashed lines). Current measurements of pressure and flows taken at the monitoring node are the input for the model (the solid lines). At each time step these measurements make up an n-dimensional vector (a test point). This vector is normalised with respect to the training set and processed according to the algorithm presented in Section 2. The model's output is a current state indication in the binary form (normal /fault).

4. Chojnice WDS case study

4.1. Experiment set-up. For the purpose of performance evaluation, a well-calibrated model of the water distribution system in the Chojnice town (northern Poland) is used (Duzinkiewicz *et al.*, 2008). The model was prepared in Epanet—a public-domain water distribution system modelling software package. It was used as a substitute of measurement equipment to generate values of flows and pressures in monitoring nodes both during correct operation and simulated faults.

Based on the documentation of the network, the entire area was split into semi-independent zones and a single residential zone was chosen for further study. Within this area three monitoring nodes were selected empirically: 030, 074 and 088 (Fig. 4(a)).

Three training sets (one for each monitoring node) were generated from Epanet's model. Each set consisted of data collected in fault-free conditions during 6 consecutive days with various demand patterns. The measurements were made every 5 minutes, and hence the training set consisted of 1728 samples.

The test set consisted of a number of different scenarios. An individual scenario contained three data sets (again, one for each monitoring node) from a single day (288 samples) with demand patterns different from the training ones and a fault (leakage of a fixed size) simulated in a random node within a selected zone at a random time. Only single-fault cases were considered.

The local KPCA models of the monitoring nodes, like all other necessary computational algorithms used for fault detection (Fig. 3), were implemented in the MATLAB environment (self-developed scripts). Parameters of σ and r were determined by an exhaustive discrete two-dimensional grid search based on results from 10 test scenarios. This allowed determining the parameters for each monitoring node ('030': $r = 70$, $\sigma = 0.8$, '074': $r = 120$, $\sigma = 1.2$, '088': $r = 130$, $\sigma = 1.5$).

Figure 4(b) presents the reconstruction error evaluated for each monitoring node for a sample test scenario. It is normalised with respect to the corresponding E^{MAX} values for better readability. For the node 074, the threshold upon fault is visibly exceeded, thus allowing detecting the leakage. Before the fault occurred, there were a few false alarms.

4.2. Performance evaluation. There are four possible outcomes for a single input vector $x_{new\,i}$ processed by

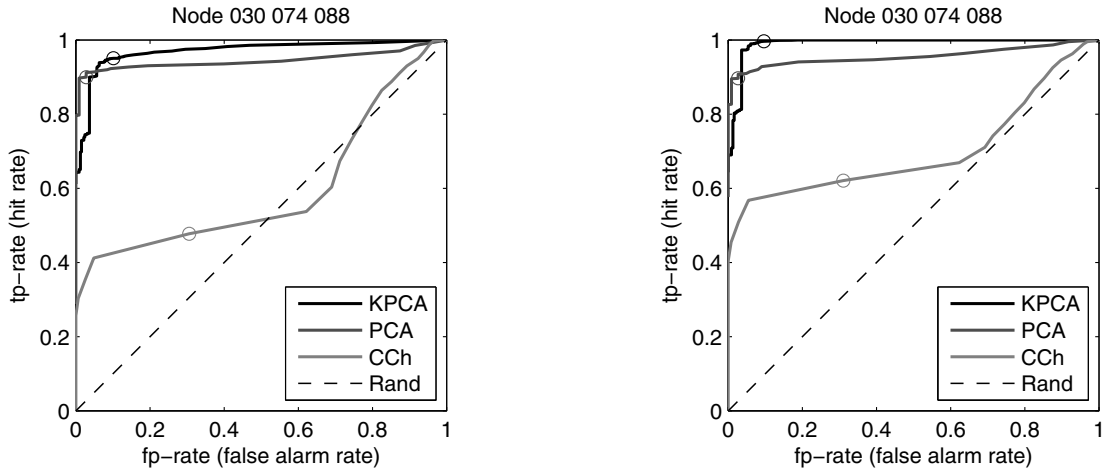


Fig. 5. ROC curves obtained from experiments: three monitoring nodes—leakage size 1.1-1.5m^3/h (a), three monitoring nodes—leakage size 1.7-2.2m^3/h (b).

a fault detection system: hit (alarm raised upon fault detection), missed alarm, normal (no alarm raised in the fault-free case), false alarm. Two of those, namely, missed alarm and false alarm, should not occur with a perfectly fine operating fault detection system. To simplify the quantitative presentation of results coming from a number of different scenarios, it is convenient to operate on rates. In this paper an average hit rate (true positive rate, tp-rate) and an average false alarm rate (false positive rate, fp-rate) are used. An ideal fault detection system should have a hit rate equal to one (detection of all faults) and a false alarm rate equal to zero (no faults indicated during fault-free operation). This is usually not possible in the case of complex real-life systems, but a good fault detection system should get as close to ideal as possible.

The tp-rate and the fp-rate allow plotting the Receiver Operating Characteristics (ROC) curve—a popular method to assess the performance of a classifier (Alpaydin, 2010). That characteristic presents the hit rate vs. the false alarms rate as the threshold changes. In order to prove the efficiency of the proposed approach, the results of experiments are given in the form of ROC curves (Fig. 5).

A single test scenario does not provide sufficient overview of fault detection capability in various operating conditions. Therefore the tp-rate and the fp-rate for each point on the characteristic are evaluated based on the average performance from 200 random test scenarios. The circle is the reference value and corresponds to the tp-rate and the fp-rate obtained using E^{MAX}, which is determined during the training phase (99th percentile over the test set). The remaining part of characteristics is drawn by re-evaluating performance in identical conditions with the value of E^{MAX} changed gradually.

For the purpose of performance comparison, two other methods were evaluated using an identical approach: PCA—following the description by Jezior *et al.* (2007), and a simple Control Chart (CCh), where a fault is indicated whenever any variable exceeds fixed upper or lower boundaries.

4.3. Overview of the results. The performance of a fault detection system consisting of three local models (node 030, 074, 088) was evaluated for two different sets of test scenarios: a simulated leakage of 1.1–1.5 m^3/h (Fig. 5(a)) and 1.7–2.2 m^3/h (Fig. 5(b)). With the daily average demand of the entire network around 180 m^3/h, this represents reasonably sized leakages. Depending on σ and r, it is possible to build either a conservative KPCA model that would loosely enclose the training set or a strict one. The model described in this section is the latter: compared with PCA, it provides a higher hit alarm rate at the cost of a slightly higher false alarm rate. Better performance is especially visible with larger leakages (Fig. 5(b) vs. Fig. 5(a)), because a precise KPCA model allows detecting leakages within a larger range and for a similar fp-rate it achieves a higher tp-rate. Notice that for those two experiments the performance of a PCA-based detection system is similar. The system based on models using a simple control chart does not provide satisfactory results.

5. Summary

The results obtained from well-designed experiments prove that the novelty detection approach based on KPCA is an efficient way to detect leakages and possibly other faults that affect water distribution in water networks.

It significantly outperforms the control chart approach, proving that independent analysis of each variable is not sufficient, and demonstrates advantages over classic multivariate analysis (PCA). Although PCA models are preferred for the cases where data follow a linear (or almost linear) relationship due to easier parameter tuning, KPCA models allow building an arbitrary smooth decision boundary that permits to model non-linear systems such as water networks. The parametric nature of the model provides flexibility—by adjusting the KPCA model it is possible to choose between sensitivity and specificity. On the other hand, choosing the parameter is not a trivial task, so one needs to employ a search strategy, which is an interesting direction for future research.

References

Aizerman, M., Braverman, E. and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition, *Automation and Remote Control* **25**: 821–837.

Alpaydin, E. (2010). *Introduction to Machine Learning*, The MIT Press, Cambridge, MA.

Barshan, E., Ghodsi, A., Azimifar, Z. and Jahromi, M.Z. (2011). Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds, *Pattern Recognition* **44**(7): 1357–1371.

Brdyś, M., Grochowski, M., Gminski, T., Konarczak, K. and Drewa, M. (2008). Hierarchical predictive control of integrated wastewater treatment systems, *Control Engineering Practice* **16**(6): 751–767.

Campbell, W.M., Campbell, J.P., Reynolds, D.A., Singer, E. and Torres-Carrasquillo, P.A. (2006). Support vector machines for speaker and language recognition, *Computer Speech and Language* **20**(2–3): 210–229.

Duzinkiewicz, K., Borowa, A., Mazur, K., Grochowski, M., Brdys, M.A. and Jezior, K. (2008). Detection and localisation in drinking water distribution networks by multiregional PCA, *Studies in Informatics and Control* **17**(2): 135–152.

Hoffman, H. (2007). Kernel PCA for novelty detection, *Pattern Recognition* **40**(3): 863–874.

Hott, K. (2008). Robust face recognition under partial occlusion based on support vector machine with local Gaussian summation kernel, *Image and Vision Computing* **26**(11): 1490–1498.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods—A survey, *Automatica* **20**(4): 387–404.

Jackson, J.E. (1991). *A User's Guide to Principal Components*, Wiley, Newark, NJ.

Jezior, K., Mazur, K., Borowa, A., Grochowski, M. and Brdys, M. A. (2007). Multiregional PCA for leakage detection and localisation in DWDS—Chojnice case study, *in* J. Korbicz, K. Patan and M. Kowal (Eds.), *Fault Diagnosis and Fault Tolerant Control*, Academic Publishing House EXIT, Warsaw, pp. 303–310.

Kulczycki, P. and Charytanowicz, M. (2010). A complete gradient clustering algorithm formed with kernel estimators, *International Journal of Applied Mathematics and Computer Science* **20**(1): 123–134, DOI: 10.2478/v10006-010-0009-3.

Li, J., Li, X. and Tao, D. (2008). KPCA for semantic object extraction in images, *Pattern Recognition* **41**(10): 3244–3250.

Lima, C.A. and Coelho, A.L. (2011). Kernel machines for epilepsy diagnosis via EEG signal classification: A comparative study, *Artificial Intelligence in Medicine* **53**(2): 83–95.

Mashford, J., Silva, D.D., Marney, D. and Burn, S. (2009). An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine, *Proceedings of the 3rd International Conference on Network and System Security, Gold Coast, Australia,* pp. 534–539.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations, *Philosophical Transactions of the Royal Society of London, Series A* **209**(441–458): 415–446.

Nogayama, T., Takahashi, H. and Muramatsu, M. (2003). Generalization of kernel PCA and automatic parameter tuning, *IEIC Technical Report* **103**(389): 43–48.

Nowicki, A. and Grochowski, M. (2011). Kernel PCA in application to leakage detection in drinking water distribution system, *in* P. Jedrzejowicz, N.T. Nguyen and K. Hoang (Eds.), *ICCCI (1)*, Lecture Notes in Computer Science, Vol. 6922, Springer, Berlin, pp. 497–506.

Patan, K. and Korbicz, J. (2012). Nonlinear model predictive control of a boiler unit: A fault tolerant control study, *International Journal of Applied Mathematics and Computer Science* **22**(1): 225–237, DOI: 10.2478/v10006-012-0017-6.

Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.-R., Rätsch, G. and Smola, A.J. (1999). Input space versus feature space in kernel-based methods, *IEEE Transactions on Neural Networks* **10**(5): 1000–1017.

Schölkopf, B., Smola, A. and Müller, K.R. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10**(5): 1299–1319.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge.

Slišković, D., Grbić, R. and Hocenski, Ž. (2011). Methods for plant data-based process modeling in soft-sensor development, *Automatika* **52**(4): 306–318.

Thornton, J., Sturm, R. and Kunkel, G. (2008). *Water Loss Control*, McGraw-Hill Companies, New York, NY.

Venkatasubramanian, V., Rengaswamy, R. and Kavuri, S. (2003a). A review of process fault detection and diagnosis, Part II: Qualitative models and search strategies, *Computers and Chemical Engineering* **27**(3): 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. and Yin, K. (2003b). A review of process fault detection and diagnosis, Part III: Process history based methods, *Computers and Chemical Engineering* **27**(3): 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S. (2003c). A review of process fault detection and diagnosis, Part I: Quantitative model-based methods, *Computers and Chemical Engineering* **27**(3): 293–311.

Xiao-Li, C. and Jiang Chao-Yuan, G.S.-Y. (2008). Leakage monitoring and locating method of water supply pipe network, *Proceedings of the 7th International Conference on Machine Learning and Cybernetics, Kunming, China*, pp. 497–506.

Xiong, H., Swamy, M. and Ahmad, M.O. (2005). Optimizing the kernel in the empirical feature space, *IEEE Transactions on Neural Networks* **16**(2): 460–474.

Adam Nowicki was born in Poland in 1986. After receiving his M.Sc. degree in control engineering in 2010 from the Electrical and Control Engineering Department at the Gdańsk University of Technology, he enrolled in a Ph.D. programme. His research interests include machine learning, fault detection and isolation, as well as soft sensors.

Michał Grochowski received his M.Sc. degree in control engineering in 2000 from the Electrical and Control Engineering Department at the Gdańsk University of Technology. In 2004 he received the Ph.D. degree in automatic control and robotics from the same university. Since 2004 he has held the position of an assistant professor at the Control Engineering Department at the Gdańsk University of Technology. His current research is focused on predictive control of complex systems, computational intelligence, fault detection and fault tolerant control.

Kazimierz Duzinkiewicz received his M.Sc. degree in control and measurement engineering in 1973 and his Ph.D. degree in control engineering in 1982 from the Electrical and Control Engineering Department at the Gdańsk University of Technology. In 2009 he received a D.Sc. degree from the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics of the AGH University of Science and Technology in Cracow. He has been employed as a university lecturer starting his work in 1973 from the position of an assistant, with the current position of an assistant professor at the Control Systems Engineering Department at the Gdańsk University of Technology. His main areas of interest include decision support and control of complex uncertain systems, robust monitoring and control, mathematical modelling of complex uncertain systems, and computational intelligence.

Received: 8 October 2011
Revised: 11 April 2012