

## Article

# Fault-Tolerant Model Predictive Control Applied to a Sewer Network

Antonio Cembellín <sup>1,\*</sup>, María J. Fuente <sup>2</sup>, Pastora Vega <sup>3</sup> and Mario Francisco <sup>1</sup>

<sup>1</sup> Computing and Automation Department, Higher Technical School of Industrial Engineering, University of Salamanca, 37700 Béjar, Spain; mfs@usal.es

<sup>2</sup> System Engineering and Automatic Control Department, School of Industrial Engineering (EII), University of Valladolid, 47011 Valladolid, Spain; mariajesus.fuente@uva.es

<sup>3</sup> Computing and Automation Department, Faculty of Science, University of Salamanca, 37008 Salamanca, Spain; pvega@usal.es

\* Correspondence: cembe@usal.es; Tel.: +34-923-408-0800 (ext. 2237)

**Abstract:** This paper presents a Fault-Tolerant Model Predictive Control (FTMPC) algorithm applied to a simulation model for sewer networks. The aim of this work is to preserve the operation of the predictive controller as much as possible, in accordance with its operational objectives, when there may be anomalies affecting the elements of the control system, mainly sensors and actuators. For this purpose, a fault detection and diagnosis system (FDD) based on a moving window principal component analysis technique (MWPCA) will be developed to provide an online fault monitoring solution for large-scale complex processes (e.g., sewer systems) with dynamically changing characteristics, and a reconfiguration algorithm for the MPC controller taking advantage of its own features such as constraint handling. Comparing the results obtained considering various types of faults, with situations of normal controlled operation and with the behavior of the sewer network when no control is applied, will allow some conclusions to be drawn at the end.

**Keywords:** fault detection and diagnosis; principal component analysis; fault-tolerant control; model predictive control; sewer systems



**Citation:** Cembellín, A.; Fuente, M.J.; Vega, P.; Francisco, M. Fault-Tolerant Model Predictive Control Applied to a Sewer Network. *Appl. Sci.* **2024**, *14*, 5359. <https://doi.org/10.3390/app14125359>

Academic Editors: Pedro Cabrera, Javier Sanchis and Eloy Irigoyen

Received: 13 May 2024

Revised: 14 June 2024

Accepted: 19 June 2024

Published: 20 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Urban drainage networks (UDN) collect and carry both urban wastewater and that which comes from precipitation to wastewater treatment plants (WWTPs) for treatment before being discharged into the environment, constituting a combined urban drainage system (CUDS). During periods of heavy rain, the residual water resulting from the mixing can overload the urban system and produce overflows (combined system overflows, CSOs) that can be harmful to the environment. To avoid CSOs, the current UDNs have retention systems capable of storing the water that reaches the network in times of intense rain and later releasing the stored volume at lower flows suitable for treatment by the WWTPs. Adequate real-time control (RTC) of the volume of water stored in the tanks can significantly improve the operation of the network to minimize the impact of CSOs [1–4].

Among the techniques used for optimal control of these systems, those that use a simplified model of the process to predict its behavior stand out. This is how Model Predictive Control (MPC) works [5]. MPCs are part of a control methodology that uses a process prediction model to calculate the manipulated variables over a future horizon to optimize a certain cost function. It is an algorithm that has been successfully implemented for several decades and has also been applied to UDNs with great success [6–9].

On the other hand, urban water treatment systems (UWS), which integrate both UDNs and WWTPs, have a high degree of interconnection and their proper functioning depends on the reliability of the equipment used, such as sensors (flowmeters, level sensors) and actuators (pumping stations, gates, valves) and communication systems. The

environmental conditions surrounding this equipment can cause its deterioration and malfunction. For this reason, it is necessary to develop Fault Tolerant Control Systems (FTCS) to maintain safe and efficient operation. In this way, a Fault-Tolerant Controller (FTC) is one that can achieve control objectives even though faults may exist, which can lead to a reduction in system performance [10–12]. Fault-tolerant control takes advantage of the physical and analytical redundancies of the system to increase its performance when an element suffers a malfunction. Furthermore, rapid detection and identification of a fault can help avoid serious and even dangerous breakdowns.

Generally, FTCS can be classified into two types: passive (PFTCS) and active (AFTCS). AFTCS react to system component faults by actively reconfiguring control actions so that stability and performance are acceptable, even if performance has degraded [11,13]. Normally, AFTCS consist of four subsystems: (1) a reconfigurable controller, (2) a fault detection and diagnosis scheme (FDD), (3) a controller reconfiguration mechanism, and (4) a command/reference governor.

Existing FDD approaches can be generally classified into two categories: (1) model-based scheme and (2) data-driven (model-free) scheme [14,15]. Data-driven schemes are divided mainly into two approaches: the multivariate statistical process control methods (MSPC) and machine learning (ML) methodologies. In the first case, the most applied methodology is the principal component analysis (PCA) [16–20]. The second approach considered is the machine learning or artificial intelligence techniques [21–25]. Furthermore, deep learning strategies have become increasingly popular in the face of complex nonlinearity and can be used for modeling, control, or management of WWTPs as can be seen in [26–31]; however, very few studies have addressed the fault detection problems in sewer networks.

For sewer systems, some methods are based on data analysis [32]. Furthermore, there are methods using closed-circuit television (CCTV) inspections and artificial intelligence to classify defects automatically [33]. Others are based on state estimation, using, for example, a Luenberger observer [34], or determination of normal operating ranges for sensor and actuators [35,36]. Often, the controller used is an MPC, so it would be a fault-tolerant model predictive control (FTCMPC) [35,36], but no work has been found in which the PCA technique has been applied to the fault detection and diagnosis in a sewer network.

Due to complex physical and chemical processes as well as changing operating conditions and the nonlinearity of the sewer networks, this technique could be applied successfully to this process.

The benchmark considered as a case study is described in [37]. The main problem of this system is the high variability of the disturbances (collected flows in each area) that affect the process. Among the whole set of data that the benchmark realistically integrates, there are time intervals of weeks in which rainfall is very low, which means that the control system, even when working properly, has little influence on the performance of the system. In this case, a fault of any sensor or actuator, even a large one, would be virtually undetectable, although it would also have little impact on the system.

Something similar occurs in the complete opposite situation: if very heavy and repeated rainfall occurs, which can saturate the sewer network, the control system will not prevent overflows in the network, which can become important even if it works properly, and if a fault occurs in any equipment, it will have little impact on the system and therefore its detection and classification will be more difficult.

These same reasons lead us to think that when any type of fault occurs, if it is not very significant, it will most likely go unnoticed. Benchmark simulation tests demonstrate this.

Consequently, it is advisable to focus on intermediate situations, i.e., situations after times of moderately high-intensity rainfall, or at longer time intervals when medium-intensity rainfall occurs, but with more continuity. It has been found that it is in these cases that the MPC controller is most useful in reducing overflows at different points in the network and in keeping the inflow to the treatment plant closer to its nominal value.

The main contribution of this work consists of the development of a real-time online FTMPC applied to the UDN system considered, consisting of three subsystems: a fault detection system based on an adaptive online PCA moving data window technique, capable of providing a real-time fault monitoring solution for the sewerage system despite the dynamically changing properties of the system; a fault diagnosis system, which will classify the detected fault through statistical calculations that will identify the variable that deviates the most from its normal behavior; and finally, a system for reconfiguring the MPC controller, taking advantage of its constraint handling capability, to try to maintain control over the whole plant, minimizing the effects of the fault. Several case studies with different disturbance profiles will be analyzed. The results have been compared with the behavior of the system without control, with the normal MPC control algorithm and with different fault situations without reconfiguration of the system.

This article is structured as follows: after an introduction, the theoretical description of mathematical algorithms that will be used begins, i.e., MPC and PCA. Afterwards, the fault detection, diagnosis and reconfiguration methodology will be detailed. The following section exposes a case study where previous methodology will be applied: first, the sewer system description is presented, and then, the MPC control algorithm and the FTC system that will be applied. Next, the results obtained in each case will be shown to finish with the conclusions of the work.

## 2. Theoretical Framework

### 2.1. Introduction

The proposed methodology is a Real-Time MPC-based Fault-Tolerant system that includes a mixed approach (model-based and data-based approach) that uses an online PCA technique for fault detection and diagnosis. In this section, the mathematical foundation of the MPC and PCA technique will be presented, as well as the statistics used to process monitoring and fault detection and diagnosis.

### 2.2. MPC Formulation

Model Predictive Control (MPC) is a control technique that calculates the control law solving an online optimal problem [5]. Control objectives are formulated as an objective function  $J$  to be minimized over a time-finite horizon  $N$ , considering constraints like a system mathematical model, actuator, or sensor limits and/or disturbances. The problem can be stated at a certain time instant  $t$  as:

$$\begin{aligned} & \min_{\mathbf{U}(t)} J(\mathbf{X}(t), \mathbf{U}(t), \mathbf{D}(t)), \text{ subject to :} \\ & \quad \hat{x}(t) = x_0 \\ & \quad \hat{x}(k+1+t) = g_1(\hat{x}(k+t), u(k+t), d(k+t)), \quad k = 0, 1, \dots, N-1 \\ & \quad g_2(\hat{x}(k+t), u(k+t), d(k+t)) \leq 0, \quad k = 0, 1, \dots, N-1 \\ & \quad x_{\min} \leq \hat{x}(k+t) \leq x_{\max}, \quad k = 0, 1, \dots, N-1 \\ & \quad u_{\min} \leq u(k+t) \leq u_{\max}, \quad k = 0, 1, \dots, N-1 \end{aligned} \quad (1)$$

where  $N$  is the time optimization horizon;  $g_1$  and  $g_2$  represent the system model;  $x_0$  is the values of the system states at instant  $t$  and  $\mathbf{X}(t)$  is the sequence of the states,  $\mathbf{U}(t)$  is the sequence of control signals and  $\mathbf{D}(t)$  is the sequence of disturbances. These sequences are calculated or estimated at instant  $t$  and extend to  $N$ .

This formulation is suitable for adding fault-tolerant properties to the controller because the MPC problem can be updated with the information provided by the FDD module [35].

### 2.3. Principal Components Analysis (PCA)

This method has two advantages: on the one hand, it allows data from a higher-dimensional space to be represented in a reduced-dimensional space, and on the other

hand, it transforms the original correlated variables into new uncorrelated variables, which facilitates the understanding of the data [16–19].

A matrix  $\mathbf{X} \in \mathfrak{R}^{n \times m}$  representing  $n$  data taken from  $m$  process variables is arranged. So that all variables are equally considered, it is necessary to normalize the data of the matrix by columns with a mean of zero and variance of one,  $\mathbf{X}_n$ . From this matrix, the covariance matrix  $\mathbf{S}$  is calculated and decomposed using singular value decomposition (SVD):

$$\mathbf{S} = \frac{1}{n-1} \mathbf{X}_n^T \mathbf{X}_n = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T, \tag{2}$$

where  $\mathbf{\Lambda} \in \mathfrak{R}^{m \times m}$  is a diagonal matrix formed by the non-negative real eigenvalues in decreasing order of  $R$  ( $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$ ) and  $\mathbf{V}$  is formed by the eigenvectors of  $\mathbf{S}$ . The principal components can be obtained as:

$$\mathbf{T} = \mathbf{X}_n \mathbf{P}, \tag{3}$$

where  $\mathbf{P}$  is formed taking the first  $a$  columns of  $\mathbf{V}$  (also called *loadings*), and  $\mathbf{T}$  are the  $a$  principal components of  $\mathbf{X}_n$ . Each of the components of new space  $\mathbf{T}$  are called *scores*.

An important aspect to consider is what criteria to apply when selecting the number of principal components  $a$ . One method is based on selecting the number of principal components until a limit of variance (CPV) is reached. CPV values between 80% and 90% are usually selected. This method has been applied in this work.

There are different statistics for monitoring a process considering the PCA by generating control charts to monitor the state of the process. The most used for this task are:

- Hotelling statistician ( $T^2$ ): For an observation vector  $x \in \mathfrak{R}^{m \times 1}$ , this index is defined as:

$$T^2 = x^T \mathbf{P} \mathbf{\Lambda}_a^{-1} \mathbf{P}^T x, \tag{4}$$

where  $\mathbf{\Lambda}_a$  has been created with the first rows and columns of  $\mathbf{\Lambda}$ , and  $a$  is the number of principal components selected.

If this index exceeds a preset threshold, it follows that the process is not operating normally, that is, a fault has occurred. The threshold can be calculated offline using historical data, as follows:

$$T_\alpha^2 = \frac{a(n^2 - 1)}{n(n - a)} F_\alpha(a, n - a), \tag{5}$$

where  $n$  is the number of samples that have been considered for the calculation of the PCA, and  $F_\alpha(a, n - a)$  is the critical value of the function  $F$  (Fisher–Snedecor  $F$  distribution) with  $n$  and  $n - a$  degrees of freedom and  $\alpha$  the significance level, which will specify the degree of false alarm commitment. Its typical values vary between 0.01 and 0.05.

- Q or SPE statistician: This index is known as a squared prediction error (SPE). It is calculated from an observation vector  $x \in \mathfrak{R}^{m \times 1}$  as:

$$Q = r^T r, \quad r = (\mathbf{I} - \mathbf{P} \mathbf{P}^T) x, \tag{6}$$

where  $r$  is the residue vector, and  $\mathbf{I}$  is the identity matrix  $m \times m$ .

As for the  $T^2$  index, a threshold can be calculated to determine when the process stops operating normally. This threshold is obtained from historical data taken offline for the PCA:

$$Q_\alpha = \frac{v}{2\mu} \chi_\alpha^2(1 - \alpha, h)^{-1}, \quad h = 2 \frac{\mu^2}{v}, \tag{7}$$

with  $\chi_\alpha^2(1 - \alpha, h)^{-1}$  being the inverse of accumulative function  $\chi^2$ ,  $\alpha$  is the tolerance index to false alarms and  $\mu$  and  $v$  are the mean and variance of  $Q$ , respectively.

If the process is operating normally, the  $Q$  index measures noise fluctuations. When an abnormal event occurs that affects the covariance of  $\mathbf{X}$ , it is detected because the  $Q$  statistic would exceed the set threshold.

The tests carried out for the system considered, introducing various types of faults that affect sensors and actuators, have shown that the  $Q$  index is more effective than the  $T^2$  for detecting abnormal operating situations due to the great variability of the disturbances that affect the process. For this reason, this statistic will be used in this work.

### 3. Proposed Method for Fault Detection, Diagnosis and Reconfiguration

When we try to apply directly the PCA-based fault detection and diagnosis techniques, as explained in the previous section, these techniques do not work due to the high variability of the data taken from the system affected by disturbances and due to the method used to calculate the tank outlet flow setpoints, as these setpoints are continuously changing depending on the wastewater stored in the tanks at any given time. Therefore, these conditions mean that the thresholds calculated offline in normal operating situations are not useful for online fault detection and diagnosis. To try to solve this problem, it will be necessary to have normal operating data generated online to calculate the threshold corresponding to a certain range of values, considering the effect of disturbances and changes in output flow references. These data will be generated by a feedforward neural network trained with normal operating data and will form a moving data window at each time that allows us to calculate an adaptive threshold for the whole simulation data set [20,38,39]. Once trained, the network will be able to generate normal operating data based on the disturbances.

The procedure to apply will be the following:

- STEP 1: Initialization: A matrix  $\mathbf{X}_p \in \mathbb{R}^{n \times m}$  formed by  $n$  data of  $m$  process variables is filled up with operating data obtained by simulation. This matrix is normalized and is going to be considered a sliding window of process data, with length  $n$ .
- STEP 2: Obtaining the  $Q$  statistic: a data vector  $x \in \mathbb{R}^{m \times 1}$  with the measurements of the system variables is formed. The  $x$  data are normalized and the PCA method is applied to this new data, and the  $Q$  statistic is calculated as indicated in Equation (6). This value is stored and updated every time a new sample is acquired, up to a maximum of  $n$  values, because each time a new sample is taken, the oldest one is discarded.
- STEP 3:  $Q_\alpha$  ( $Q$  threshold) calculation: to obtain the threshold of the  $Q$  statistic adaptively in the sliding window, the value of the actual disturbances taken from the plant and the value of the rest of the process variables under normal operating conditions are required. These variables are calculated with a neural network. This network has as inputs the disturbances, and the past value of the process variables. The output of the network is the prediction of the process variables at the next time instant. Thus, for the online procedure, a matrix  $\mathbf{X} \in \mathbb{R}^{n \times m}$  is now constructed with the current disturbances and the network outputs, in the sliding window, to calculate the threshold of  $Q$  according to Equation (7). Since each time a new sample is taken, the oldest one is discarded, the calculated threshold is adapted to new disturbances.
- STEP 4: Fault detection: the number of times that the  $Q$  statistic exceeds the calculated threshold (number of alarms) is calculated, and if this number reaches  $M$  consecutive values, it is considered that a fault has occurred. Then, go to STEP 5; otherwise, go to STEP 2.
- STEP 5: Fault diagnosis: the fault instant is determined, and to calculate the contribution of each variable to the fault, the residue of the first  $H$  samples of  $M$  values that were used to detect the fault is evaluated by Equation (6). The variable whose mean of the residue is higher than those calculated for the  $H$  samples considered is determined and the failing device is identified.

STEP 6: Reconfiguration of the MPC: mainly, sensor and actuator faults are considered. Thus, if a sensor fails, its real value can be estimated using other measurements of the process, taking advantage of redundancy. On the other hand, if an actuator fails, MPC can be recalculated by adding new constraints about this device.

Section 4.3 will specify the details necessary for the application of this methodology to the system under consideration.

#### 4. Case Study: Benchmark Simulation Model for Sewer Networks

##### 4.1. Benchmark Model Description

Figure 1 shows the sewer system used as a benchmark to test control algorithms [37]. It is made up of six wastewater and rainwater collection areas (numbered 1 to 6 in the figure), six wastewater storage tanks (ST1, . . . , ST6, one of them, ST5, is off-line), wastewater pipes, five valves and a pump for flow control, and a wastewater treatment plant (WWTP). It involves collecting all the wastewater and conveying it to the treatment plant, maintaining a supply flow rate with the least possible variability and as close to its nominal value. This is achieved by retaining the volume collected in the tanks during heavy rainfall and by releasing that volume during drought times.

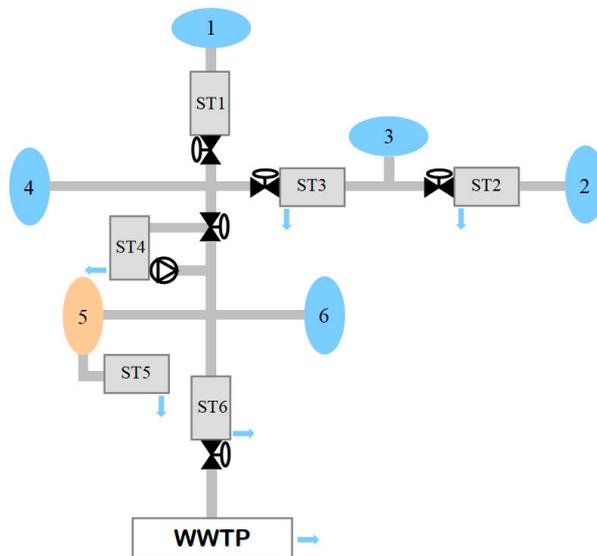


Figure 1. Catchment and sewer system.

As part of the control algorithm, a simplified model of the process has been developed that will be used to predict the behavior of the system during a given time interval [40]. The simplified mathematical model of the process is made up of the following elements:

- WATER COLLECTION AREA: all water collected in the area constitutes an inflow to the system that is treated as a disturbance.
- LINK ELEMENTS: they are wastewater conductions by gravity in open channels. Its discrete mathematical model would be the following:

$$q_i(k + 1) = \left(1 - \frac{T}{\tau_i}\right)q_i(k) + \left(\frac{T}{\tau_i}\right)q_{u,i}(k), \quad j = 1, 2, 3 \dots 9, \quad (8)$$

where:

$q_{u,i}(k)$  is the sum of inflows to the link element  $i$

$q_i(k)$  is the output flow of the element  $i$

$\tau_i$  is the time constant of the element  $i$

$T$  is the sampling period

- STORAGE TANKS: these are places where wastewater is stored. Its discrete model is the following:

$$\begin{aligned} h_i(k+1) &= h_i(k) + \frac{T}{A_i} [u_{in,i}(k) - u_i(k) - q_{ov,i}(k)] \\ u_i(k) &= a_i(k)c_{0i}\sqrt{h_i(k)}, \quad V_i(k) = A_i h_i(k), \quad V_{\max,i} = A_i h_{\max,i}, \quad j = 1, 2, \dots, 5, \end{aligned} \quad (9)$$

where all parameters are related to tank  $i$  and instant  $k$ :

$u_{in,i}(k)$  is the inlet flow rate

$u_i(k)$  is the outlet flow rate

$q_{ov,i}(k)$  is the overflow flow rate

$V_{\max,i}$  is the maximum capacity of the tank

$V_i(k)$  is the volume stored at the instant  $k$

$c_{0i}$  is the discharge coefficient calculated empirically for each reservoir  $i$

$A_i$  is the surface of the base of the tank  $i$

$h_{\max,i}$  is the tank height  $i$

$h_i(k)$  is the water level  $i$

$a_i(k)$  is the opening of the deposit  $i$  outlet gate (control variable:  $a_i \in [0, 1]$ )

- NODES: they represent places of confluence of several wastewater pipes. The resulting flow is the sum of the tributary flows:

$$u_{in1} + u_{in2} + \dots + u_{inm} = u_{out}, \quad j = 1, 2, 3, \dots, 9, \quad (10)$$

The states considered are the levels of the tanks ST1, ST2, ST3, ST4 and ST6 ( $x_1, \dots, x_5$ ) and the output flows of the link elements that communicate the tanks ST2 and ST3, ST4 and ST6 and the ST6 with the WWTP, which correspond to the states ( $x_6, \dots, x_9$ ). The output flows of the link elements of the water collection zones 1, 2, 4, 5 and 6 and the flow collected in zone 3, will be considered as measurable disturbances in the process: ( $d_1, \dots, d_6$ ). The system inputs are the desired flows at the outlet of each of the tanks (manipulated variables): ( $u_1, \dots, u_5$ ). The model appears in detail in [40], where the equations of the linearized model of the system that is going to be used as a prediction model in the MPC algorithm are shown:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_p\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k), \quad \text{and } \mathbf{B} = [\mathbf{B}_p \quad \mathbf{B}_d], \quad (11)$$

where:

$$\begin{aligned} \mathbf{x} &= (h_1, h_2, h_3, h_4, h_5, q_3, q_7, q_8, q_9), \\ \mathbf{u} &= (u_1, u_2, u_3, u_4, u_5), \quad \mathbf{d} = (q_1, q_2, q_4, q_5, q_6, q_{r3}), \end{aligned} \quad (12)$$

Possible faults that can affect the system operation, usually happen in sensors (level meters, flowmeters) or actuators (gates controlling the tanks output flow rates). Thus, the study of system faults will focus on these pieces of equipment.

#### 4.2. Model Predictive Control Algorithm

The control objective is to ensure a flow rate in the treatment plant that maximizes its capacity without exceeding a maximum value, avoiding overflows in the tanks and in the station itself as much as possible, and minimizing operating costs.

The algorithm used for predictive control uses a linear model of the process in the state space for prediction that includes disturbances and a cost function that calculates the outlet flow rates of the tanks for optimal operation. These flow rates calculated by the MPC are the setpoints for local output flow regulation loops for each tank, type P-I.

The cost function or objective function of the MPC is a quadratic form that considers both the tracking errors in the states, as well as the increases in the control sequence with respect to the flow reference (penalizing the control efforts), if the prediction and control horizons coincide, and their value is  $N$  [5]:

$$J(k, \mathbf{U}) = \sum_{i=1}^{N-1} \left[ \|\hat{\mathbf{x}}(k+i) - \mathbf{x}_{ref}(k)\|_{\mathbf{Q}(k)}^2 + \|\mathbf{u}(k+i) - \mathbf{u}_{ref}(k)\|_{\mathbf{R}}^2 \right] + \mathbf{x}^T(k+N)\mathbf{P}\mathbf{x}(k+N), \quad (13)$$

where:

$$\mathbf{x}_{ref}(k) = (x_{1ref}, x_{2ref}, \dots, x_{16ref}), \quad \mathbf{u}_{ref}(k) = (u_{1ref}, u_{2ref}, \dots, u_{5ref}), \quad (14)$$

The optimization problem that MPC solves is:

$$\mathbf{U}^*(k) = \underset{\mathbf{U}(k)}{\operatorname{argmin}} J(k, \mathbf{U}), \quad \mathbf{U}^*(k) = [\mathbf{u}^*(k) \quad \mathbf{u}^*(k+1) \quad \dots \quad \mathbf{u}^*(k+N-1)]^T \quad (15)$$

subject to:

$$\begin{aligned} \hat{\mathbf{x}}(k+i+1) &= \mathbf{A}(k)\hat{\mathbf{x}}(k+i) + \mathbf{B}_p(k)\mathbf{u}(k+i) + \mathbf{B}_d(k)\mathbf{d}(k+i), \quad \mathbf{d}(k+i) = \mathbf{d}(k) \\ \hat{\mathbf{x}}(k) &= \mathbf{x}(k) \\ 0 &\leq \hat{x}_j(k+i), \quad i = 0, \dots, N-1, \quad j = 1, \dots, 5 \\ 0 &\leq \hat{x}_j(k+i) \leq q_{\max j}, \quad i = 0, \dots, N-1, \quad j = 6, \dots, 9 \\ 0 &\leq u_j(k+i) \leq u_{\max j}, \quad i = 0, \dots, N-1, \quad j = 1, \dots, 5, \end{aligned} \quad (16)$$

where  $q_{\max j}$  and  $u_{\max j}$  are the upper bounds for flow rate in the link elements and the tank outputs, respectively.

The matrices  $\mathbf{Q}(k)$ ,  $\mathbf{P}$  and  $\mathbf{R}$  serve to penalize the tracking errors and the control efforts (inputs) and will be used as controller tuning parameters, as will the control horizon  $N$  ( $\mathbf{P}$  is a terminal penalty for MPC stability obtained by means of the Riccati equation [5]).

A variable  $\mathbf{Q}(k)$  matrix has been considered so that if an overflow occurs in a tank, the corresponding weight is modified to avoid it as much as possible. The non-zero elements of  $\mathbf{Q}(k)$  that constitute MPC tuning parameters are  $q_9$  and:

$$q_i = \begin{cases} f_i & x_i \leq h_{\max i} \\ f_i (1 + \alpha_i (x_i - h_{\max i})^2) & x_i > h_{\max i} \end{cases}, \quad j = 1, 2, \dots, 5, \quad (17)$$

Parameters  $f_j$  y  $\alpha_i$  are used to tune the MPC algorithm.

To achieve optimal system operation, the level setpoints are calculated following the strategy of distributing the current volume of water among all tanks as evenly as possible considering their capacity. This is achieved by calculating for each one its reference level as a function of the total capacity of the network and the capacity of that tank [40]:

$$x_{iref}(k) = \left( \frac{V_G(k)}{A_i} \right) v_i, \quad j = 1, 2, \dots, 5, \quad (18)$$

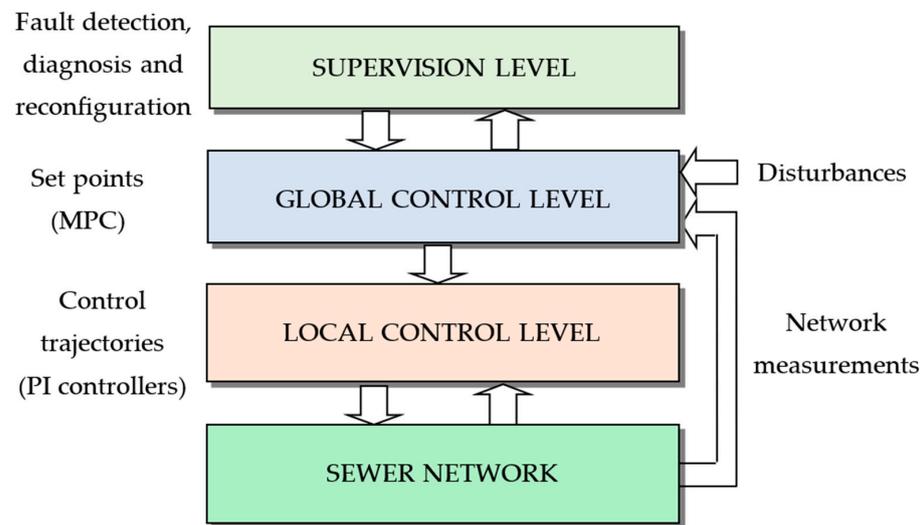
where  $x_{iref}(k)$  is the reference level for deposit  $i$  at time  $k$ ;  $V_G(k)$  is the total volume occupied at the moment  $k$ ;  $v_i$  is a factor that represents the weight of the tank capacity  $i$  in the total available volume (sum of all tank volumes);  $V_i$  and  $A_i$  are the maximum capacity and the surface area of tank  $i$ , respectively. The reference values for the flow rates would be zero because they are not considered except  $x_{ref9} = 60.000 \text{ m}^3/\text{d}$ , which is the desired inlet flow for the WWTP.

The  $\mathbf{R}$  matrix is a matrix whose first five values of the diagonal are used to penalize the variations in the flow references with respect to their reference value and the rest are null since they cannot be optimized because they correspond to the disturbances. The flow references are calculated using the desired level for each tank according to Equation (19) for the first five values:

$$u_{iref}(k) = c_{0i} \sqrt{x_{iref}(k)}, \quad j = 1, 2, \dots, 5, \quad (19)$$

#### 4.3. Fault-Tolerant Control System for the Sewer Network

A Fault-Tolerant Control system (FTC) will be implemented following the architecture corresponding to a hierarchical control system, as shown in Figure 2:



**Figure 2.** Hierarchical fault-tolerant control system.

As already indicated, the MPC generates the setpoints of the local P-I type regulators by solving a Quadratic Programming (QP) optimization problem considering the measurements of the system variables (levels and flows), the measurable disturbances of the process (inlet flows to the sewer network that comes from urban wastewater and precipitation). At the supervision level is the fault detection and diagnosis system (FDD), which receives the measurements of the system variables and disturbances from the level immediately below and allows the MPC algorithm of that level to be reconfigured after a fault has been detected and diagnosed correctly.

To reduce the number of cases in this study, only the possibility of faults in actuators (gates that regulate the flow rate of the tanks) and level sensors of each tank of the network will be considered. The performance of the fault detection system will be studied considering different moments when a fault will occur in the controlled system and two different disturbance scenarios. The faults under study will be:

- Faults in the level sensors of each tank: the system behavior will be analyzed considering faults in the sensor gain, which is reduced to 10% of its nominal value.
- Faults in the actuators: the behavior of the detection and diagnosis system will be studied considering the gate of each tank blocked at 20% of its total opening.

Both types of faults are of high magnitude, justified by the reasons explained in the introduction. These faults will occur on the second, fifth or eighth day of the total 10 days for different simulation scenarios. The methodology explained in Section 3 will be applied to the sewer system considering the following parameter values:

- The length of the sliding window is  $n = 50$  because it has been heuristically proven that 50 samples are a suitable value. Matrix  $\mathbf{X}_p \in \mathbb{R}^{n \times m}$  contains these samples.
- The data vector  $x \in \mathbb{R}^{m \times 1}$ , taken from the system, includes the system disturbances, the system state variables (tank levels and flow rates of the link elements 6, 7, 8 and 9), the output flows and output flow setpoints of each tank, so its length is  $m = 25$ .
- A matrix  $\mathbf{X} \in \mathbb{R}^{n \times m}$  is created with normal operating data using the neural network, including disturbances, and, considering a variance percentage of 95%, a fault threshold in that interval of 50 samples is calculated for the  $Q$  statistic.
- The number of consecutive alarms considered to be a fault is  $M = 20$ . This is a suitable value to avoid false detections caused by the strong disturbances affecting the system.

- For diagnosis, the residue of the first 10 samples of the set of 20 that were used to detect the fault is evaluated ( $H = 10$ ), as explained in Section 3. This value has been chosen experimentally. The variable whose mean of the residue is greater than those calculated for the set of 10 samples considered is determined. If this variable is the level of a specific tank, it is considered that the corresponding level sensor fails. If the variable is an outlet flow rate, it follows that the correspondent reservoir gate fails.
- The MPC reconfiguration depends on the element that presents the fault, so different strategies will be applied to reconfigure the MPC controller to minimize the effects caused by the faults:
- Faults in level sensors: when a level sensor fails, the level value of that tank  $h_i(k)$  can be estimated at instant  $k$  if its output flow  $u_i(k)$ , its discharge coefficient  $c_{0i}$  and the gate opening degree  $v_i(k)$  are known by Equation (20):

$$h_i(k) = \left[ \frac{u_i(k)}{c_{0i}v_i(k)} \right]^2, \quad 0 < v_i \leq 1, \quad i = 1, 2, \dots, 5, \tag{20}$$

- Faults in gates: in this case, the MPC algorithm is reconfigured by adding to the QP optimization problem an equality constraint for the calculation of the reference output flow rate of the tank affected,  $u_{iref}(k)$  since its output flow  $u_i(k)$  can be measured at instant  $k$ , so:

$$u_{iref}(k) = u_i(k), \quad i = 1, 2, \dots, 5, \tag{21}$$

To have normal online performance data, a neural network trained with fault-free data extracted from the benchmark will be used. The neural network is composed of 25 neurons in the hidden layer and 19 in the output layer, as shown in Figure 3. To improve the performance of the neural network, it is fed back with the system outputs from the previous instant. Thus, the disturbances ( $d_1, \dots, d_6$ ) and the previous outputs are applied to the input ( $i_1, \dots, i_{25}$ ), and the process variables of interest are obtained at the output ( $o_1, \dots, o_{19}$ ): state variables ( $x_1, \dots, x_9$ ), flow references ( $u_{ref1}, \dots, u_{ref5}$ ) and output flow rates of every tank ( $u_1, \dots, u_5$ ).

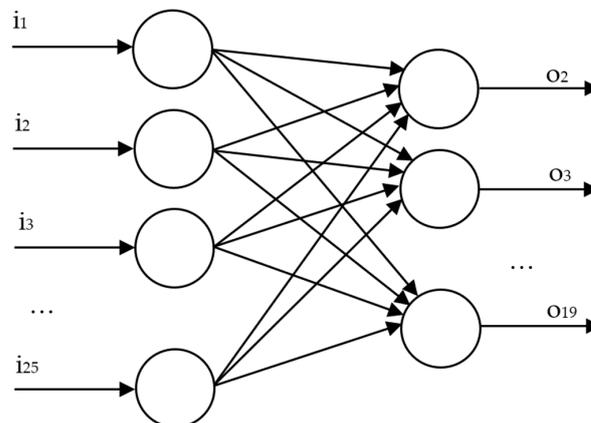


Figure 3. Neural network structure.

### 5. Results and Discussion

Three scenarios have been considered, extracted from the data time-series of the benchmark, in which the flow variations are more significant according to the reasons explained in the introduction of the article. The first scenario provides the neural network training data that will be used to generate normal online operating data. The second and third scenarios will serve to evaluate the fault detection and diagnosis system, as well as its reconfiguration.

To perform the simulation tests, the weights of the MPC cost function (non-null elements of matrices  $Q(k)$  and  $R$ ) have been adjusted for Equation (17) and are shown in Table 1:

**Table 1.** MPC cost function weights.

$Q(k)$		$R$
$f_i = 1, i = 1, 2, 3$	$\alpha_i = 10, i = 2, 3$	$r_i = 10^{-8}, i = 1, \dots, 5$
$f_4 = 10$	$\alpha_4 = 100$	
$f_5 = 50$	$\alpha_5 = 200$	
$q_9 = 10^{-6}$		

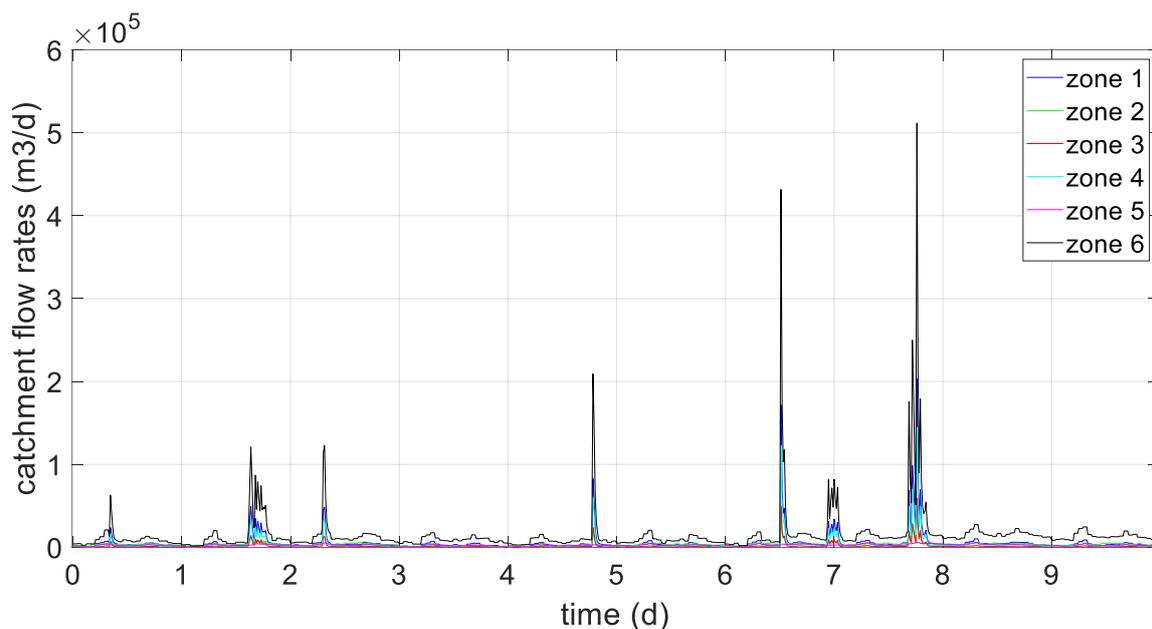
The model system parameters are shown in Table 2:

**Table 2.** System parameters.

Parameter	Units	Values
$A_1, \dots, A_5$ tank areas	$m^2$	1188, 252, 348, 852, 2988
$c_{01}, \dots, c_{05}$ discharge coefficients	$m^{5/2}/d$	1.89, 0.40, 0.55, 1.36, 6.12 ( $\times 10^4$ )
$h_{\max 1}, \dots, h_{\max 5}$ tank heights	m	5 (for all)
$h_{\min 1}, \dots, h_{\min 5}$ minimum levels	m	0 (for all)
$q_{\max 1}, \dots, q_{\max 9}$ maximum flow rates at the pipes outlet	$m^3/d$	5.99, 1.27, 3.02, 4.29, 4.29, 15.06, 4.29, 23.64, 6 ( $\times 10^4$ )
$T$ sampling time	d	0.0021
$\tau_1, \dots, \tau_9$ link elements time constants	d	0.0313, 0.0104, 0.0104, 0.0208, 0.0208, 0.073, 0.0208, 0.0104, 0.0104
$u_{\max 1}, \dots, u_{\max 5}$ maximum flow rates at the reservoirs outlet	$m^3/d$	5.98, 1.27, 1.75, 4.29, 19.34 ( $\times 10^4$ )

### 5.1. Training and Validation of the Neural Network

Figure 4 represents the inlet flows to the sewer network collected in each of the catchment areas considered, due to precipitation and wastewater, over a period of 10 days (scenario 1). It shows the training data profile of the neural network extracted from the benchmark:



**Figure 4.** Catchment flow rates (scenario 1).

Then, to validate the trained network, the data provided by the system and the neural network will be compared with input data corresponding to scenarios 2 and 3, shown in Figures 5 and 6, respectively:

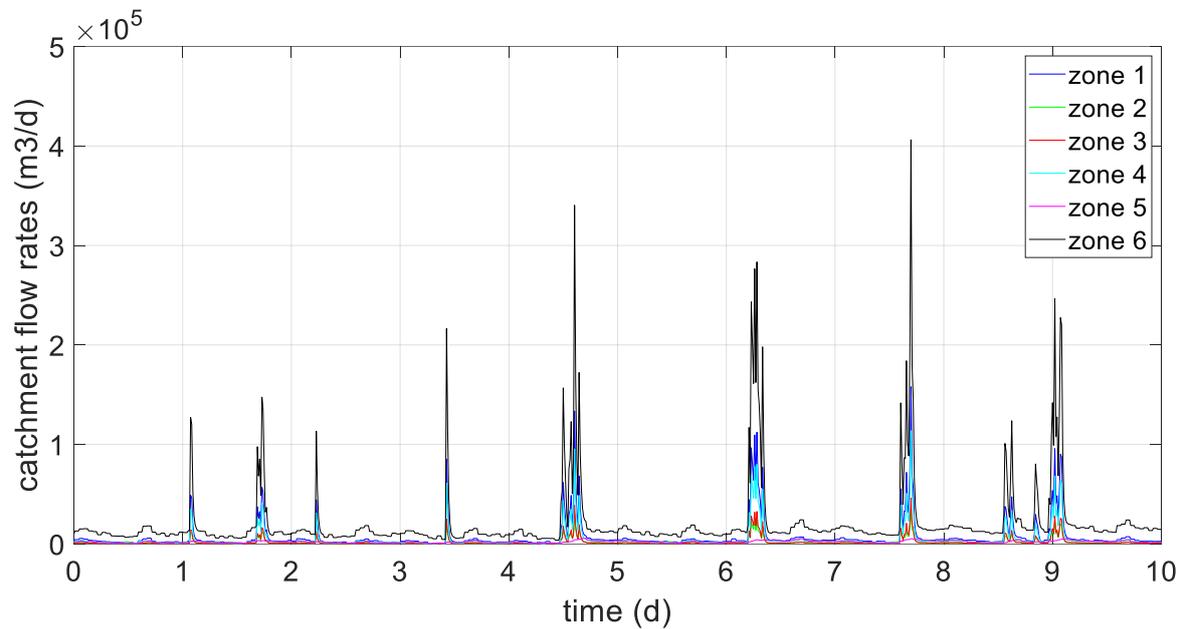


Figure 5. Catchment flow rates (scenario 2).

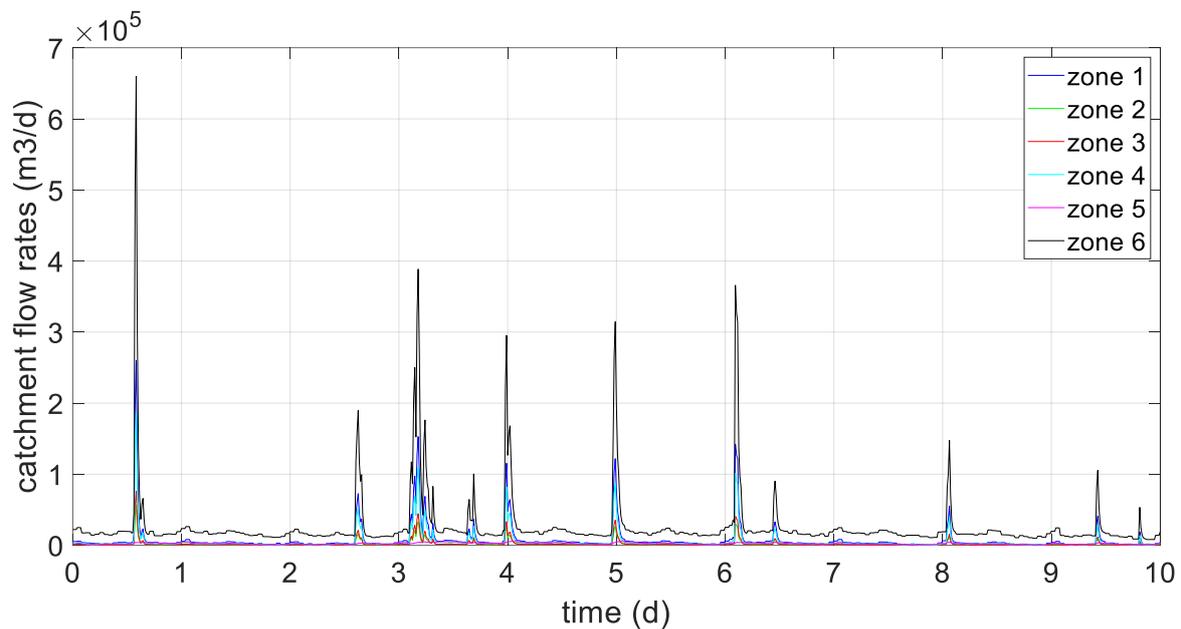


Figure 6. Catchment flow rates (scenario 3).

Below are some results related to the evaluation of the trained network. Figures 7 and 8 show the reservoir levels provided by the system and by the neural network under normal operating conditions with input data from scenario 2. Similarly, Figures 9 and 10 depict the same levels with input data from scenario 3.

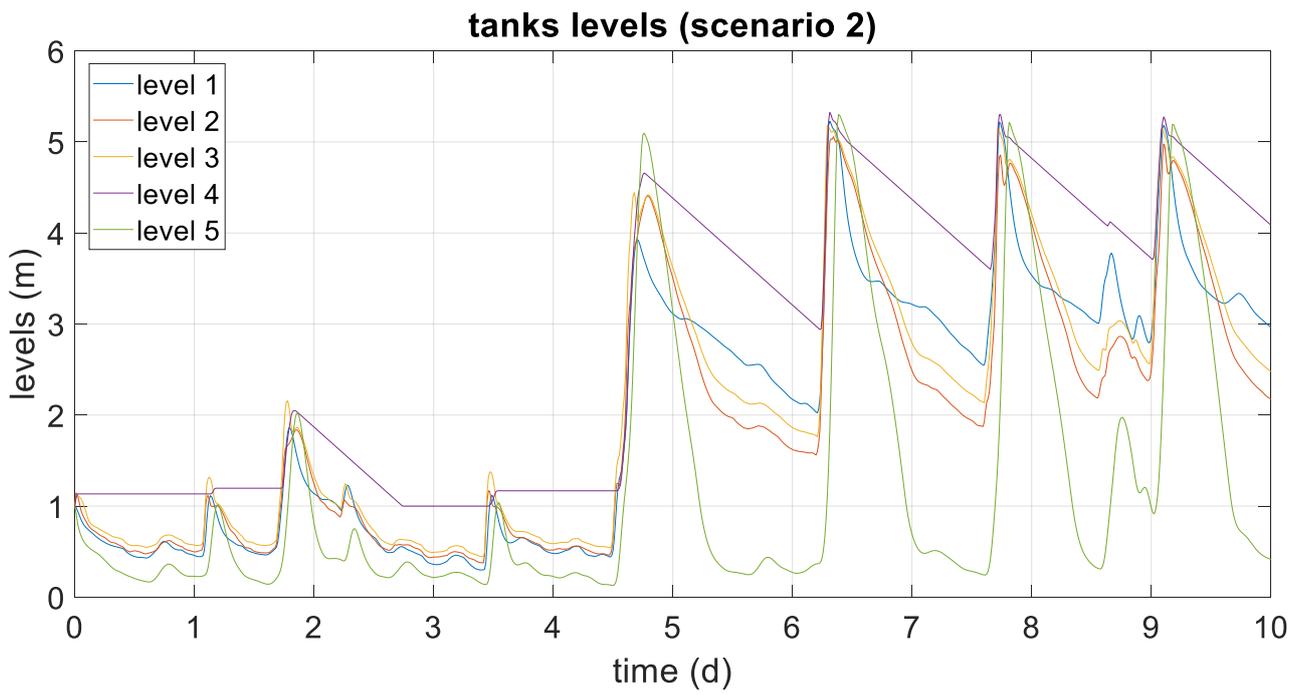


Figure 7. Tank levels (scenario 2).

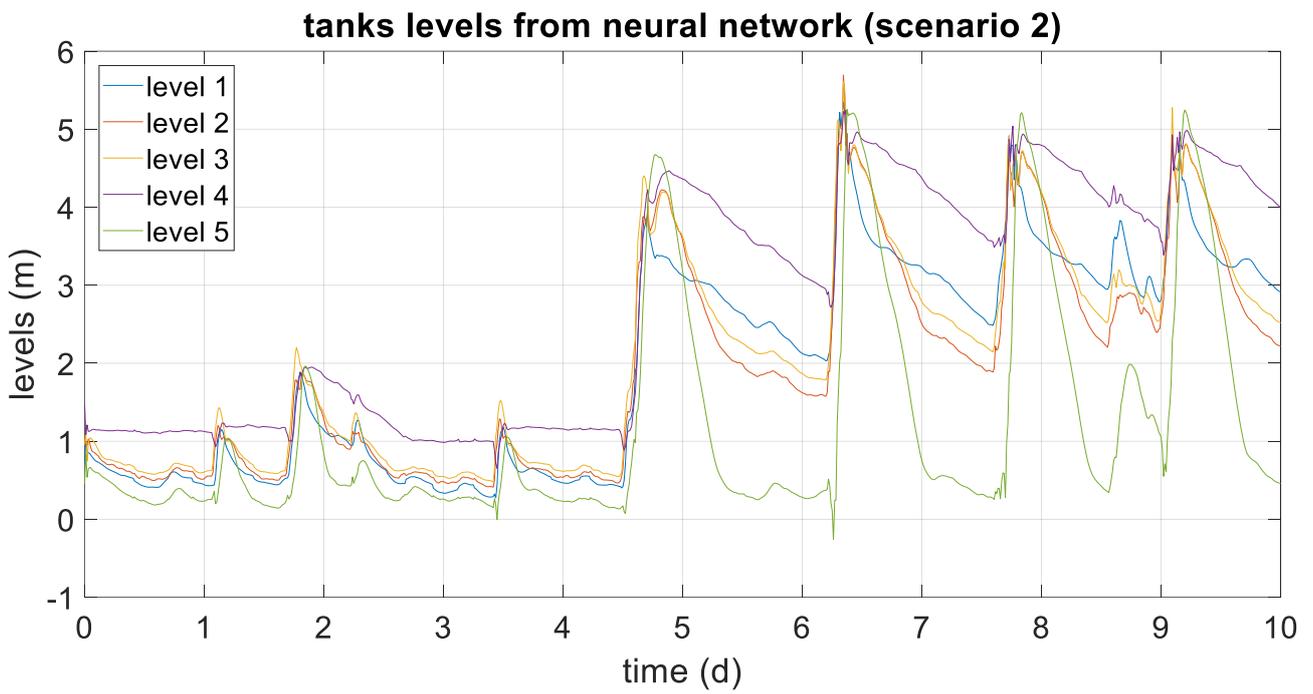


Figure 8. Tank levels generated by the neural network (scenario 2).

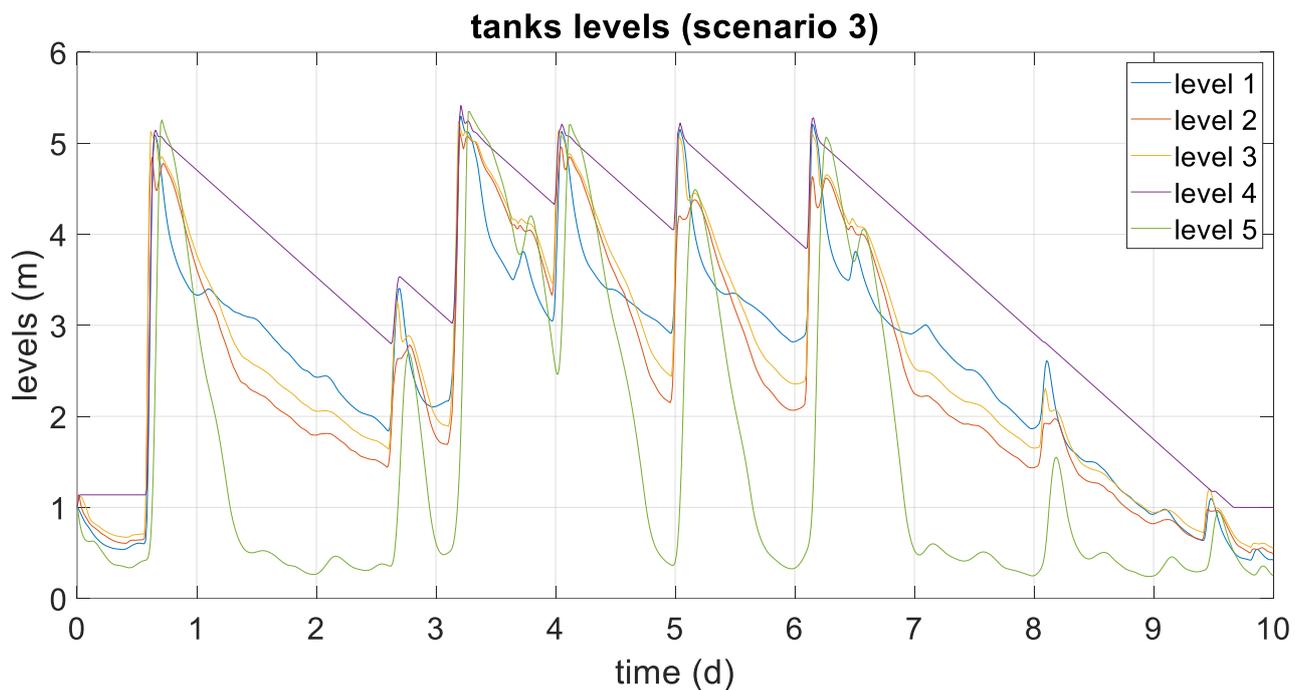


Figure 9. Tank levels (scenario 3).

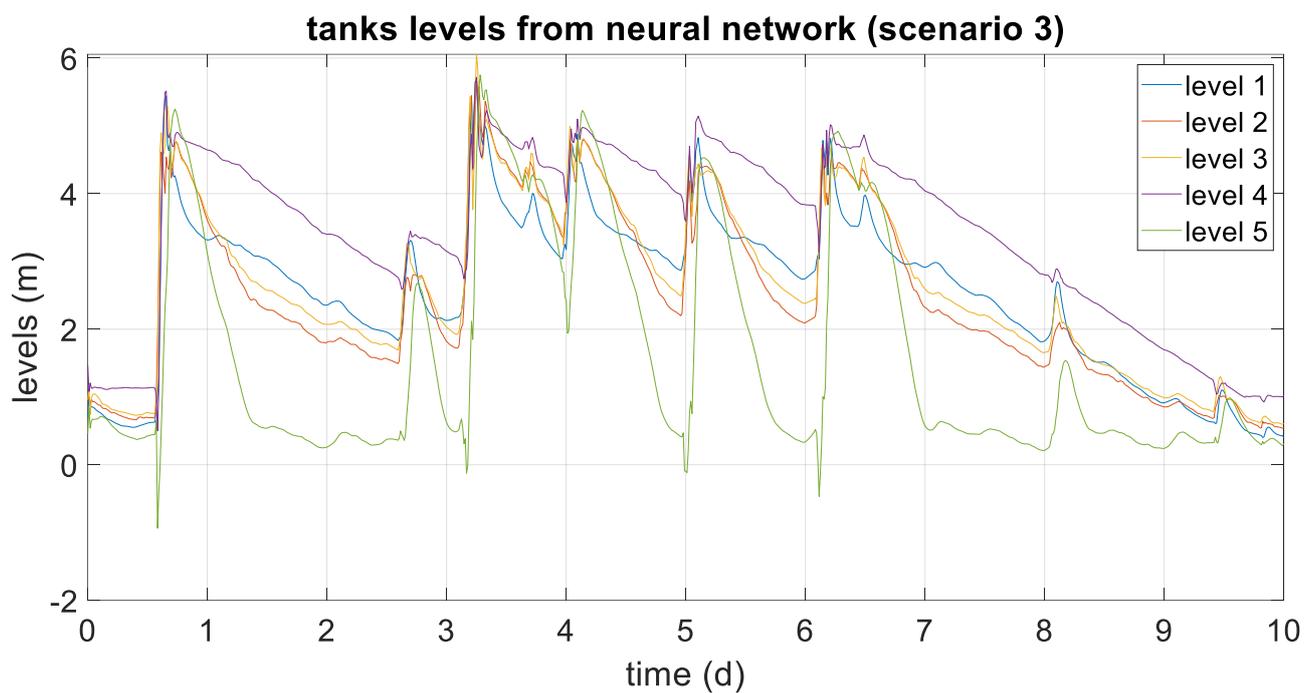


Figure 10. Tank levels generated by the neural network (scenario 3).

It is found that the results provided by the network in both cases largely match those generated by the system, so the network can be used to generate normal operating data based on the disturbances affecting the system.

### 5.2. Fault Detection and Diagnosis Tests

As exposed in Section 4.3, the faults under study will be:

- Faults in level sensors: sensor gain is reduced to 10% of its nominal value.

- Faults in actuators: gate is blocked at 20% of its total opening.

These faults will be provoked on the second, fifth or eighth day of a 10-day simulation interval, considering both scenarios 2 and 3.

Previously, it has been verified that in the absence of faults, the detection system does not detect any type of fault, although it presents an alarm rate for scenarios 2 and 3 of 9.57% and 11.13%, respectively, but since 20 consecutive alarms are necessary to consider a fault situation, none are detected.

The following graphics show the calculated  $Q$  threshold and the  $Q$  statistic in the absence of faults for scenarios 2 and 3:

Both Figures 11 and 12 show that the  $Q$  statistic sometimes exceeds the calculated threshold, but no fault is detected because 20 consecutive alarms are necessary.

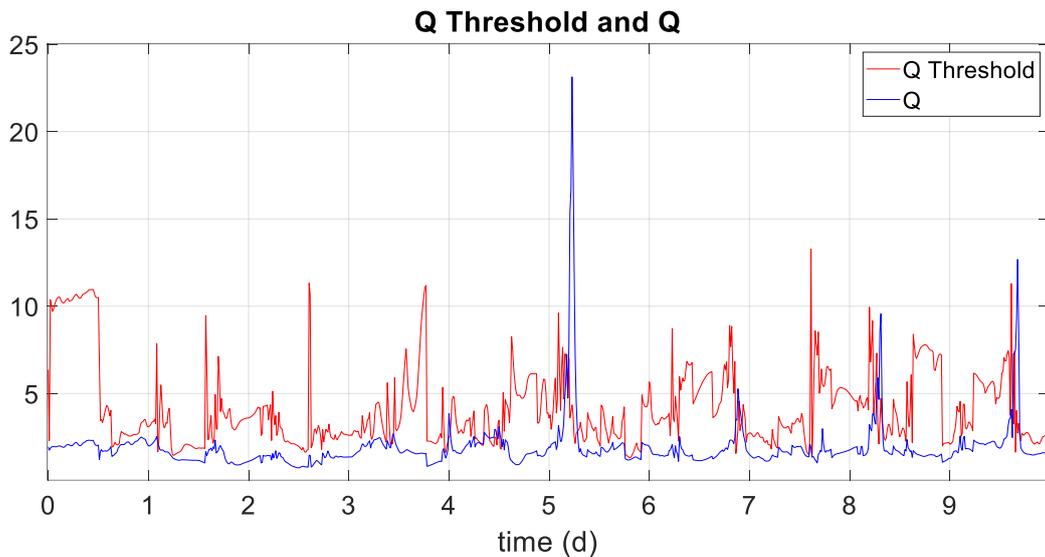


Figure 11.  $Q$  statistic and threshold calculated in a normal operating situation (scenario 2).

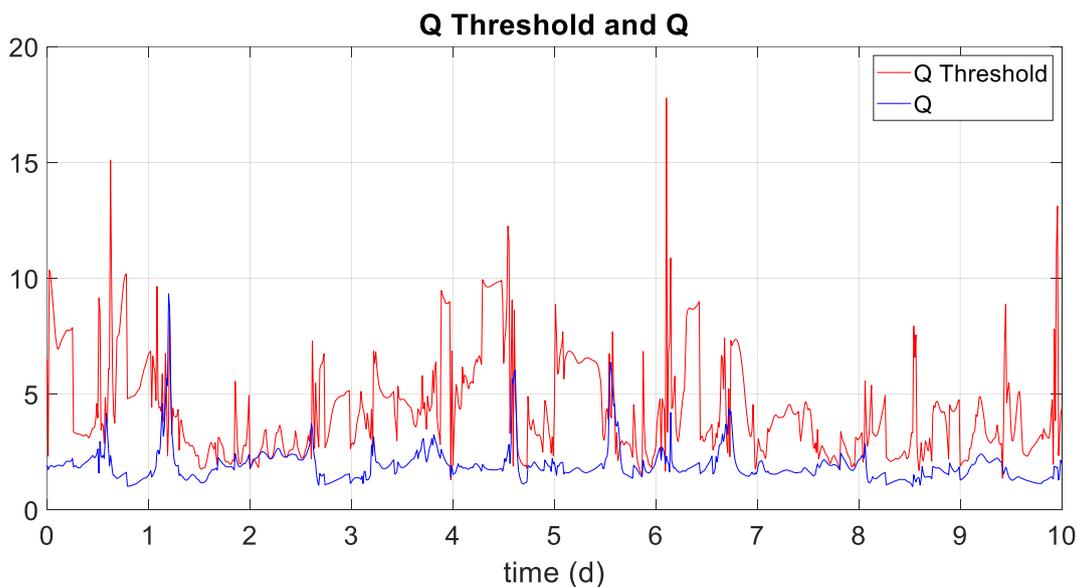


Figure 12.  $Q$  statistic and threshold calculated in a normal operating situation (scenario 3).

The results obtained in different fault situations are shown in Tables 3 and 4 for scenarios 2 and 3. The tables show the detection results (detection instant) and the diagnosis

for the type of fault considered (fault variable:  $h_i$ , is the tank  $i$  level;  $u_i$  is the tank  $i$  output flow rate). The correct diagnosis is highlighted in green:

**Table 3.** Fault detection and diagnosis results for scenario 2.

Fault Type	Fault on the 2nd Day		Fault on the 5th Day		Fault on the 8th Day	
	Detection (Day)	Diagnosis (Variable)	Detection (Day)	Diagnosis (Variable)	Detection (Day)	Diagnosis (Variable)
Level sensor 1	2.285	h1	5.26	h1	8.26	h1
Level sensor 2	2.521	h2	5.26	u2	8.26	h2
Level sensor 3	2.625	u3	5.26	h3	8.271	h3
Level sensor 4	2.75	h4	5.812	u1	8.26	h4
Level sensor 5	3.083	h2	5.521	h5	8.521	h5
Gate 1	2.438	u3	5.312	h5	8.464	h5
Gate 2	2.346	u3	5.277	u2	8.327	u2
Gate 3	2.31	u3	5.269	u3	8.317	u3
Gate 4	5.346	u4	5.531	u1	8.562	u1
Gate 5	2.49	h4	5.531	u1	8.344	u3

**Table 4.** Fault detection and diagnosis results for scenario 3.

Fault Type	Fault on the 2nd Day		Fault on the 5th Day		Fault on the 8th Day	
	Detection (Day)	Diagnosis (Variable)	Detection (Day)	Diagnosis (Variable)	Detection (Day)	Diagnosis (Variable)
Level sensor 1	2.26	h1	5.26	u1	8.26	h1
Level sensor 2	2.194	u1	5.26	h3	8.26	h2
Level sensor 3	2.26	h3	5.31	h3	8.26	h3
Level sensor 4	2.865	h4	5.26	h4	8.76	u4
Level sensor 5	2.302	h5	5.594	h5	8.896	h2
Gate 1	2.26	u1	5.365	u1	8.579	u1
Gate 2	3.146	u2	5.537	u3	8.419	h3
Gate 3	2.271	u3	5.485	u3	8.387	u3
Gate 4	2.219	u1	5.735	u1	8.677	u4
Gate 5	2.26	u5	5.604	u3	8.438	h4

In terms of fault detection, both scenarios show that all faults are detected relatively quickly (almost all before the next day). It must be considered, as already mentioned, that these are large faults. In the tests performed with less significant faults, detection was considerably delayed with respect to the time of fault generation, and there were some cases in which the fault was not detected at all.

Regarding fault classification, success is variable depending on the proximity of the disturbances to the moment of generation of the fault, as well as their magnitude and their frequency. As these characteristics are highly variable, success in detecting the fault is also variable. Further investigations must be improved by fault classification.

### 5.3. Fault Detection, Diagnosis and MPC Reconfiguration Tests

In this section, to assess the MPC reconfiguration performance, comparative results of the control system in four cases will be shown:

- Case 1: sewer network without control, that is, always with all the gates open.
- Case 2: sewer network controlled with MPC in the absence of faults.

- Case 3: sewer network controlled with MPC in the presence of a certain fault.
- Case 4: sewer controlled with reconfigured MPC (FTMPC). Once the fault is correctly detected and identified, the controller is reconfigured to improve system performance compared to the previous case.

In each case, scenarios 2 and 3 will be considered. Two of the most representative faults have been selected:

- Fault in the tank 1 level sensor, in which its gain is reduced to 10% of its normal value.
- Fault in the tank 3 gate, which is supposed to be blocked at 20% of its total opening.

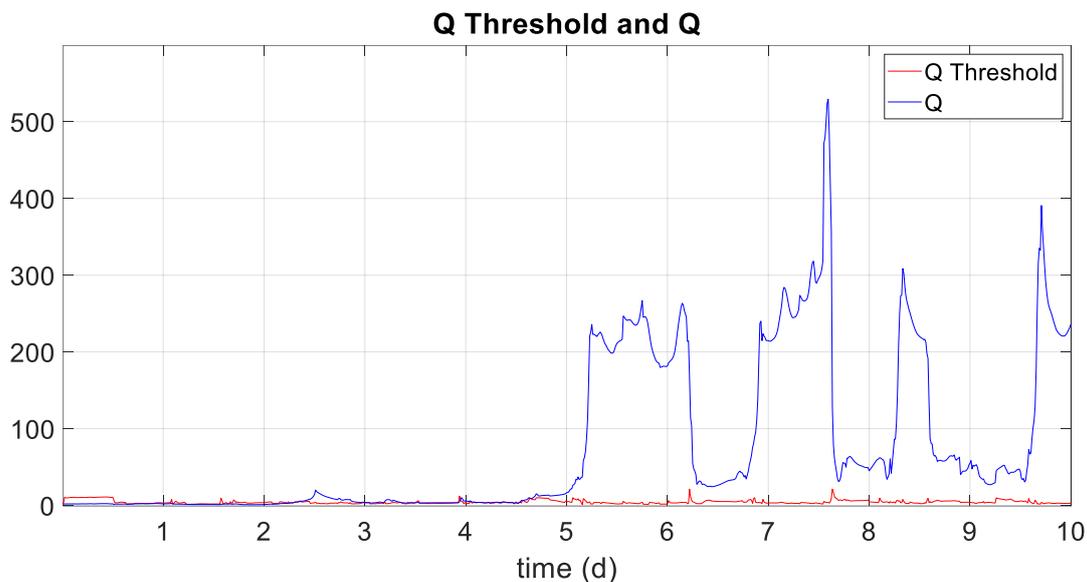
Furthermore, to better evaluate the effect of the fault and the reconfiguration of the system, it will be assumed that, in all cases, the fault is generated on the second day of the 10-day simulation period considered for each scenario.

The performance evaluation criteria will be the same as those detailed in [40]. In summary, these criteria are number of overflows ( $N_{ov}$ ), duration of overflow ( $T_{ov}$ ) in *days*, volume overflowed ( $V_{ov}$ ) in  $m^3$ , degree of utilization of WWTP ( $G_u$ ) in %, and smoothness in the application of control signals ( $S$ ) in  $m^3/d$ .

### 5.3.1. Scenario 2 Results

- Fault in the tank 1 level sensor: alarm percentage before a fault detection: 2.2%. Detection instant: 2.285 days. MPC controller is reconfigured using Equation (20) to estimate tank 1 level assuming tank 1 outlet flow rate measure is correct.

Figure 13 shows the  $Q$  threshold and the  $Q$  statistic calculated online for fault detection. Detection occurs when  $Q$  exceeds the threshold 20 consecutive times.



**Figure 13.**  $Q$  statistic and threshold calculated in level fault situation (scenario 2).

The following table provides the comparative data of system performance, including normal operating situation, fault without reconfiguration, and fault with controller reconfiguration (FTMPC). Note that the sewer configuration causes the overflowed volume in tank 1,  $V_{ov,1}$  returns to the network, and for this reason, it is not added to  $V_{ov}$  in all tables [40].

For simplicity, the main indices to be considered are  $V_{ov}$ ,  $G_u$  and  $S$ . As can be seen in Table 5, a normal MPC controller offers the best performance because the total overflow is the lowest as well as the smoothness in the control actions and provides the highest degree of utilization of the WWTP. MPC with the fault considered reduces the system performance, worsening all indices, but by comparing with no control case, its performance is much better since  $G_u$  is 57.92% vs. 53.96%,  $V_{ov}$  is  $3.8003 \times 10^4$  vs.  $6.8473 \times 10^4$  ( $m^3$ ).

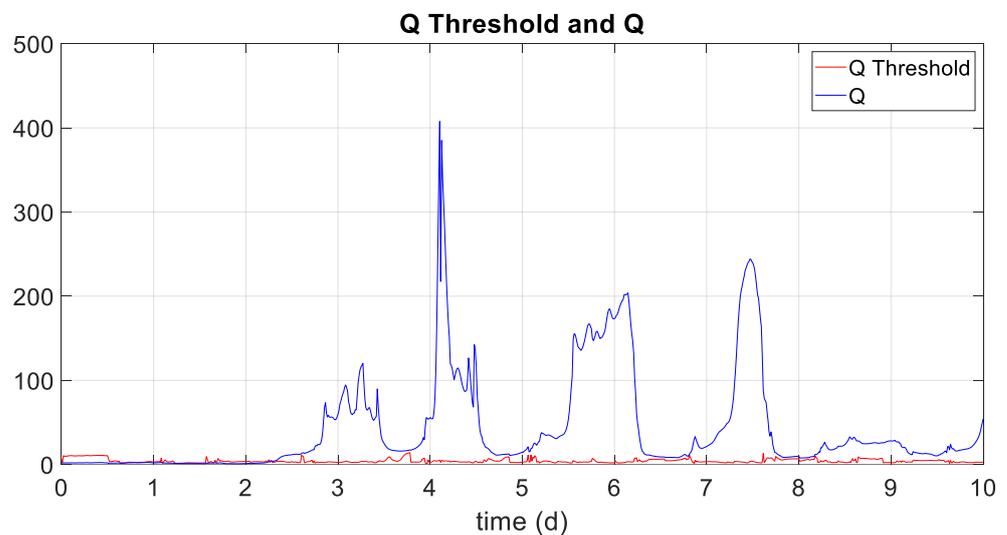
**Table 5.** System performance in every case: fault in tank 1 level sensor (scenario 2).

Data	No Control	Normal MPC	MPC with h1 Fault	Reconfigured MPC
$N_{ov,1}$	0	3	1	1
$N_{ov,2}$	0	1	1	1
$N_{ov,3}$	1	3	3	3
$N_{ov,4}$	3	3	4	3
$N_{ov,5}$	1	4	4	4
$N_{ov,WWTP}$	6	5	6	5
$T_{ov,1}$	0	0.0396	1.0917	0.7417
$T_{ov,2}$	0	0.0146	0.0063	0.0104
$T_{ov,3}$	0.0146	0.0437	0.0396	0.0396
$T_{ov,4}$	0.0771	0.0917	0.1167	0.0917
$T_{ov,5}$	0.0083	0.0604	0.0583	0.0583
$T_{ov,WWTP}$	0.3563	0.5854	0.5333	0.4937
$V_{ov,1}$	0	$5.3998 \times 10^3$	$5.0830 \times 10^4$	$3.5204 \times 10^4$
$V_{ov,2}$	0	83.5089	47.0085	79.0783
$V_{ov,3}$	556.6261	$1.8417 \times 10^3$	$1.6621 \times 10^3$	$1.6522 \times 10^3$
$V_{ov,4}$	$7.9044 \times 10^3$	$1.3097 \times 10^4$	$2.3892 \times 10^4$	$1.8166 \times 10^4$
$V_{ov,5}$	314.0450	$1.2763 \times 10^4$	$1.2246 \times 10^4$	$1.3402 \times 10^4$
$V_{ov,WWTP}$	$5.9698 \times 10^4$	148.6536	155.0684	$1.2201 \times 10^3$
$V_{ov}$	$6.8473 \times 10^4$	$2.7935 \times 10^4$	$3.8003 \times 10^4$	$3.4520 \times 10^4$
$Q_{WWTP}$	$3.2379 \times 10^4$	$3.5974 \times 10^4$	$3.4753 \times 10^4$	$3.5129 \times 10^4$
$G_H$	53.9648	59.9560	57.9210	58.5486
$S$	-	$6.8276 \times 10^{10}$	$6.7398 \times 10^{10}$	$3.1712 \times 10^{11}$

Finally, by comparing an MPC with a fault with FTCMPC, this one improves the system performance since the total overflow volume is reduced from  $3.8003 \times 10^4$  to  $3.4520 \times 10^4$  (m<sup>3</sup>) and the degree of WWTP utilization increases from 57.92% to 58.55%, although  $S$  is worse because the system needs greater control efforts. Therefore, this reconfiguration strategy improves system performance when this fault occurs.

- Fault in the tank 3 gate: alarm percentage before a fault detection: 2.01%. Detection instant: 2.31 days. MPC controller is reconfigured by Equation (21) by adding a new constraint to the MPC problem, assuming the tank 3 outlet flow rate measure is correct.

Figure 14 shows the  $Q$  threshold calculated online and the value of the  $Q$  statistic for fault detection. Detection happens when  $Q$  exceeds the threshold 20 consecutive times. Table 6 provides the comparative data of system performance in each case for scenario 2.



**Figure 14.**  $Q$  statistic and threshold calculated in gate fault situation (scenario 2).

**Table 6.** System performance in every case: fault in tank 3 gate (scenario 2).

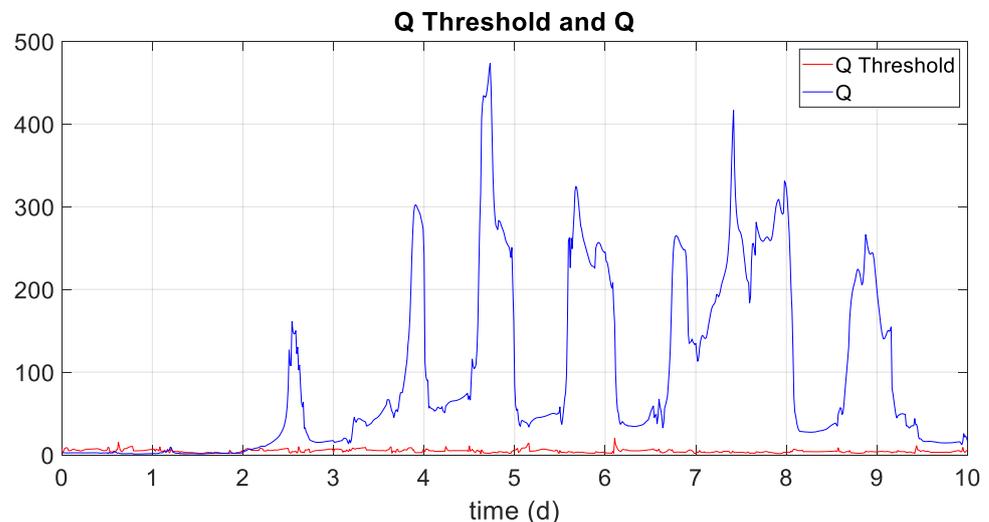
Data	No Control	Normal MPC	MPC with u3 Fault	Reconfigured MPC
$N_{ov,1}$	0	3	3	3
$N_{ov,2}$	0	1	3	3
$N_{ov,3}$	1	3	4	4
$N_{ov,4}$	3	3	3	3
$N_{ov,5}$	1	4	4	3
$N_{ov,WWTP}$	6	5	6	6
$T_{ov,1}$	0	0.0396	0.0396	0.0375
$T_{ov,2}$	0	0.0146	0.0354	0.0396
$T_{ov,3}$	0.0146	0.0437	0.6896	0.6896
$T_{ov,4}$	0.0771	0.0917	0.0792	0.0813
$T_{ov,5}$	0.0083	0.0604	0.0563	0.0500
$T_{ov,WWTP}$	0.3563	0.5854	0.5271	0.5250
$V_{ov,1}$	0	$5.3998 \times 10^3$	$4.6476 \times 10^3$	$4.1425 \times 10^3$
$V_{ov,2}$	0	83.5089	716.7745	907.9697
$V_{ov,3}$	556.6261	$1.8417 \times 10^3$	$1.6621 \times 10^3$	$1.6522 \times 10^3$
$V_{ov,4}$	$7.9044 \times 10^3$	$1.3097 \times 10^4$	$9.1273 \times 10^3$	$9.2190 \times 10^3$
$V_{ov,5}$	314.0450	$1.2763 \times 10^4$	$1.1609 \times 10^4$	$1.1453 \times 10^4$
$V_{ov,WWTP}$	$5.9698 \times 10^4$	148.6536	134.7366	134.5941
$V_{ov}$	$6.8473 \times 10^4$	$2.7935 \times 10^4$	$3.5235 \times 10^4$	$3.5162 \times 10^4$
$Q_{WWTP}$	$3.2379 \times 10^4$	$3.5974 \times 10^4$	$3.5092 \times 10^4$	$3.5114 \times 10^4$
$G_u$	53.9648	59.9560	58.4874	58.5230
$S$	-	$6.8276 \times 10^{10}$	$6.7277 \times 10^{10}$	$6.7843 \times 10^{10}$

Considering the main indices ( $V_{ov}$ ,  $G_u$  and  $S$ ), the results discussion is like the previous section, with the normal MPC controller having the best performance. Furthermore, MPC with the fault considered reduces the system performance, but by comparing with no control case, its performance is much better since  $G_u$  is 58.49% vs. 53.96%,  $V_{ov}$  is  $3.5235 \times 10^4$  vs.  $6.8473 \times 10^4$  ( $m^3$ ). Finally, by comparing the MPC with a fault with FTCMPC, this one improves the system performance slightly since the total overflow volume is reduced from  $3.5235 \times 10^4$  to  $3.5162 \times 10^4$  and the degree of WWTP utilization increases from 58.49% to 58.52%, but  $S$  is worse because the system needs higher control efforts.

5.3.2. Scenario 3 Results

- Fault in the tank 1 level sensor: alarm percentage before a fault detection: 2.44%. Detection instant: 2.26 days. MPC reconfiguration is performed in the same way as in Section 5.3.1 for the tank 1 level sensor.

Figure 15 shows the  $Q$  threshold and the value of the  $Q$  statistic calculated online. Detection occurs when  $Q$  exceeds the threshold 20 consecutive times.



**Figure 15.**  $Q$  statistic and threshold calculated in level fault situation (scenario 3).

Table 7 provides the comparative data of system performance in every case. These results allow to achieve the same conclusions as the scenario 2 results:

**Table 7.** System performance in every case: fault in tank 1 level sensor (scenario 3).

Data	No Control	Normal MPC	MPC with h1 Fault	Reconfigured MPC
$N_{ov,1}$	0	5	2	2
$N_{ov,2}$	0	1	1	1
$N_{ov,3}$	2	5	4	5
$N_{ov,4}$	5	5	5	5
$N_{ov,5}$	1	4	4	4
$N_{ov,WWTP}$	7	6	6	6
$T_{ov,1}$	0	0.0604	1.4750	0.8292
$T_{ov,2}$	0	0.0250	0.0125	0.0125
$T_{ov,3}$	0.0396	0.0729	0.0583	0.0583
$T_{ov,4}$	0.1271	0.1333	0.1292	0.1250
$T_{ov,5}$	0.0187	0.0813	0.0688	0.0688
$T_{ov,WWTP}$	0.4562	0.7708	0.7250	0.7333
$V_{ov,1}$	0	$6.9554 \times 10^3$	$5.6633 \times 10^4$	$4.0860 \times 10^4$
$V_{ov,2}$	0	385.1159	150.8754	214.8527
$V_{ov,3}$	$1.1834 \times 10^3$	$2.5109 \times 10^3$	$2.0415 \times 10^3$	$2.2327 \times 10^3$
$V_{ov,4}$	$1.1874 \times 10^4$	$1.8044 \times 10^4$	$2.7861 \times 10^4$	$2.4317 \times 10^4$
$V_{ov,5}$	$3.2433 \times 10^3$	$1.9535 \times 10^4$	$1.7412 \times 10^4$	$1.7102 \times 10^4$
$V_{ov,WWTP}$	$7.1094 \times 10^4$	219.6231	222.5543	$213.9156 \times 10^3$
$V_{ov}$	$8.7395 \times 10^4$	$4.0695 \times 10^4$	$4.7688 \times 10^4$	$4.4080 \times 10^4$
$Q_{WWTP}$	$3.7001 \times 10^4$	$4.1641 \times 10^4$	$4.0370 \times 10^4$	$4.1286 \times 10^4$
$G_u$	61.6680	69.4022	67.2835	68.8092
$S$	-	$6.3711 \times 10^{10}$	$6.4402 \times 10^{10}$	$1.0547 \times 10^{11}$

Regarding the main indices ( $V_{ov}$ ,  $G_u$  and  $S$ ), the results discussion is like previous cases, with the normal MPC controller having the best performance. Furthermore, the MPC with the fault considered reduces the system performance, but by comparing with no control case, its performance is better since  $G_u$  is 67.28% vs. 61.67%,  $V_{ov}$  is  $4.77 \times 10^4$  vs.  $8.74 \times 10^4$  (m<sup>3</sup>). Furthermore, by comparing the MPC with a fault with FTCMPC, this one improves the system performance slightly since the total overflow volume is reduced from  $4.77 \times 10^4$  to  $4.41 \times 10^4$  and the degree of WWTP utilization increases from 67.28% to 68.81%, but  $S$  is worse because the system needs greater control efforts.

- Fault in the tank 3 gate: alarm percentage before a fault detection: 3.05%. Detection instant: 2.271 days. MPC reconfiguration is performed in the same way as in Section 5.3.1 for the tank 3 gate.

Figure 16 shows the  $Q$  threshold calculated online and the value of the  $Q$  statistic for fault detection and allows knowing the detection instant. Detection occurs when  $Q$  exceeds the threshold 20 consecutive times.

Table 8 presents the comparative data of system performance in each case. Looking at the main indicators, you can see the same behavior as in previous cases. For instance, by comparison of an MPC with a fault with FTCMPC,  $G_u$  is 67.28% vs. 68.81% and  $V_{ov}$  is  $4.6066 \times 10^4$  vs.  $4.5926 \times 10^4$  (m<sup>3</sup>). Therefore, FTCMPC improves the performance of the system.

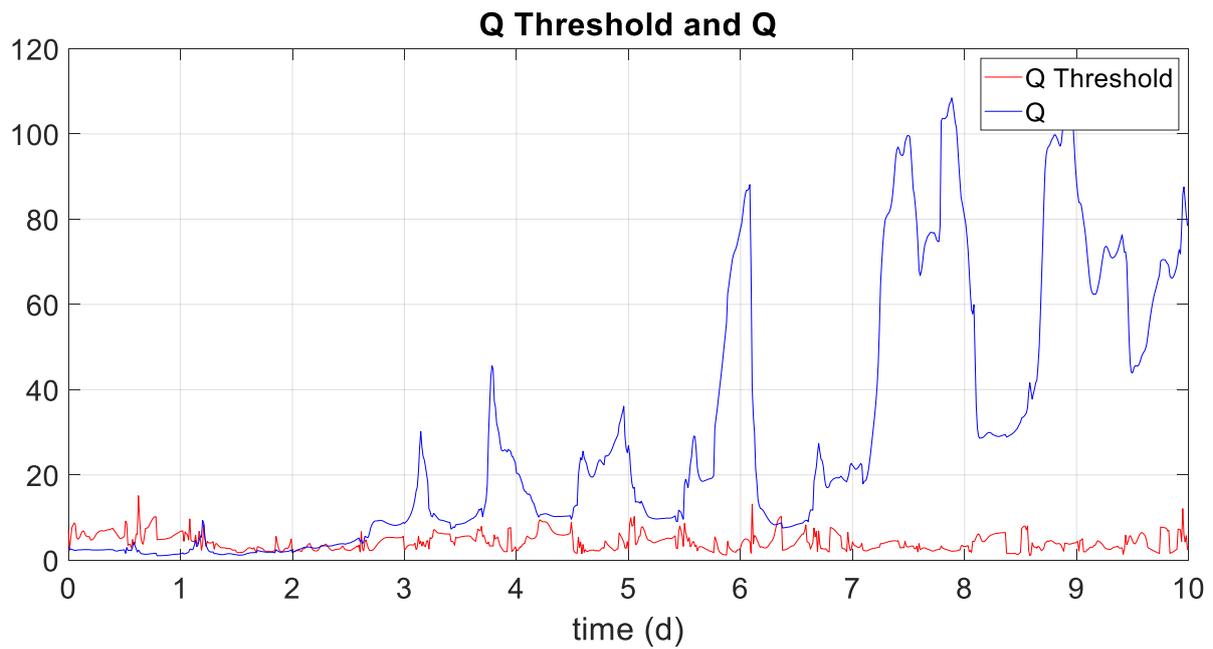


Figure 16. Q statistic and threshold calculated in gate fault situation (scenario 3).

Table 8. System performance in every case: fault in tank 3 gate (scenario 3).

Data	No Control	Normal MPC	MPC with u3 Fault	Reconfigured MPC
$N_{ov,1}$	0	5	5	5
$N_{ov,2}$	0	1	4	4
$N_{ov,3}$	2	5	7	7
$N_{ov,4}$	5	5	5	5
$N_{ov,5}$	1	4	4	4
$N_{ov,WWTP}$	7	6	6	6
$T_{ov,1}$	0	0.0604	0.0604	0.0542
$T_{ov,2}$	0	0.0250	0.0500	0.0542
$T_{ov,3}$	0.0396	0.0729	0.6708	0.6667
$T_{ov,4}$	0.1271	0.1333	0.1167	0.1187
$T_{ov,5}$	0.0187	0.0813	0.0750	0.0750
$T_{ov,WWTP}$	0.4562	0.7708	0.7354	0.7354
$V_{ov,1}$	0	$6.9554 \times 10^3$	$6.4800 \times 10^3$	$5.6858 \times 10^3$
$V_{ov,2}$	0	385.1159	$1.0779 \times 10^3$	$1.4689 \times 10^3$
$V_{ov,3}$	$1.1834 \times 10^3$	$2.5109 \times 10^3$	$1.3599 \times 10^4$	$1.3222 \times 10^4$
$V_{ov,4}$	$1.1874 \times 10^4$	$1.8044 \times 10^4$	$1.3507 \times 10^4$	$1.3430 \times 10^4$
$V_{ov,5}$	$3.2433 \times 10^3$	$1.9535 \times 10^4$	$1.7682 \times 10^4$	$1.7603 \times 10^4$
$V_{ov,WWTP}$	$7.1094 \times 10^4$	219.6231	200.5045	$201.2536 \times 10^3$
$V_{ov}$	$8.7395 \times 10^4$	$4.0695 \times 10^4$	$4.6066 \times 10^4$	$4.5926 \times 10^4$
$Q_{WWTP}$	$3.7001 \times 10^4$	$4.1641 \times 10^4$	$4.0910 \times 10^4$	$4.0938 \times 10^4$
$G_u$	61.6680	69.4022	67.2835	68.8092
S	-	$6.3711 \times 10^{10}$	$6.3251 \times 10^{10}$	$6.3165 \times 10^{10}$

## 6. Conclusions

In this paper, a methodology for fault detection and diagnosis in certain types of sensors and actuators of a wastewater sewer network, based on an adaptive PCA technique, has been presented and analyzed. Due to the peculiar characteristics of the system, subject to strong disturbances of high variability, only large faults have been detected and classified since low-intensity faults do not affect the performance of the system very much and, therefore, their detection is more difficult. Even so, the detection algorithm used has managed to detect faults in different elements and in different scenarios, with some speed and reliability. Regarding the diagnosis of the detected faults, its classification is very difficult, not only because of the disturbances in the system but also because the set point of the flow regulators is constantly being recalculated; therefore, the results obtained can be improved and it is intended to continue in this sense with the work started. For cases in which both detection and diagnosis have been successful, the MPC reconfiguration strategies show an improvement in system performance compared to that obtained when a malfunction situation occurs, and the controller is not acted upon. On the other hand, by its own structure, MPC facilitates the reconfiguration process when a fault occurs, for instance, by adding a new constraint to the optimization problem. Therefore, MPC reconfiguration is usually easy to implement, and many systems combine both strategies FTC and MPC, using a FTCMPC. Finally, although this FTCMPC controller has been designed for a sewer system, it can be easily adapted to other types of processes that present the same difficulties.

**Author Contributions:** M.J.F. and P.V. conceived and designed the general approach; A.C. and M.J.F. performed the simulations; M.J.F., P.V. and M.F. contributed to the simulation interpretations; writing—review and editing, A.C., M.J.F., P.V. and M.F.; supervision, M.J.F., P.V. and M.F.; A.C. wrote the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Spanish Government through the MICCIN/AEI/10.13039/501100011033 projects PID2019-105434RB-C31, TED2021-129201B-I00 and PID2019-105434RB-C32.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Kroll, S.; Weemaes, M.; Van Impe, J.; Willems, P. A Methodology for the Design of RTC Strategies for Combined Sewer Networks. *Water* **2018**, *10*, 1675. [[CrossRef](#)]
2. Marinaki, M.; Papageorgiou, M. *Optimal Real-Time Control of Sewer Networks*; Springer: Secaucus, NJ, USA, 2005.
3. Barbu, M.; Vilanova, R. Fuzzy Control Applied on a Benchmark Simulation Model for Sewer Networks. In Proceedings of the 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 13–15 October 2016.
4. García, L.; Barreiro-Gómez, J.; Escobar, E.; Téllez, D.; Quijano, N.; Ocampo-Martínez, C. Modeling and Real-Time Control of Urban Drainage Systems: A review. *Adv. Water Resour.* **2015**, *85*, 120–132. [[CrossRef](#)]
5. Rawlings, J.; Mayne, D. *Model Predictive Control: Theory and Design*; Nob Hill Publishing: Madison, WI, USA, 2017.
6. Sun, C.; Bernat, J.D.; Cembrano, G.; Puig, V.; Meseguer, J. Advanced Integrated Real-Time Control of Combined Urban Drainage Systems using MPC. In Proceedings of the 13th International Conference on Hydroinformatics, Palermo, Italy, 1–6 July 2018.
7. Ocampo-Martínez, C.; Puig, V.; Cembrano, G.; Quevedo, J. Application of predictive control strategies to the management of complex networks in the urban water cycle. *IEEE Control Syst. Mag.* **2013**, *33*, 15–41. [[CrossRef](#)]
8. Meneses, E.J.; Gaussens, M.; Jakobsen, C.; Mikkelsen, P.S.; Grum, M.; Vezzaro, L. Coordinating Rule-Based and System-Wide Model Predictive Control Strategies to Reduce Storage Expansion of Combined Urban Drainage Systems: The Case Study of Lundtofte, Denmark. *Water* **2018**, *10*, 76. [[CrossRef](#)]
9. Svensen, J.L.; Niemann, H.H.; Poulsen, N.K. Model Predictive Control of Overflow in Sewer Networks: A comparison of two methods. In Proceedings of the 2019 4th Conference of Control and Fault Tolerant Systems (SysTol), Casablanca, Morocco, 18–20 September 2019; pp. 412–417. [[CrossRef](#)]
10. Blanke, M.; Kinnaert, M.; Lunze, J.; Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*, 2nd ed.; Springer: Berlin, Germany, 2006.

11. Zhang, Y.; Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annu. Rev. Control* **2008**, *32*, 229–252. [[CrossRef](#)]
12. Lu, Y.; Liang, J.; Zhuang, W.; Yin, G.; Feng, J.; Zhou, C. Four-Wheel Independent Drive Vehicle Fault Tolerant Strategy Using Stochastic Model Predictive Control with Model Parameter Uncertainties. *IEEE Trans. Veh. Technol.* **2024**, *73*, 3287–3299. [[CrossRef](#)]
13. Abbaspour, A.; Mokhtari, S.; Sargolzaei, A.; Yen, K.K. A Survey on Active Fault-Tolerant Control Systems. *Electronics* **2020**, *9*, 1513. [[CrossRef](#)]
14. Ge, Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometr. Intell. Lab. Syst.* **2017**, *171*, 16–25. [[CrossRef](#)]
15. Jiang, Q.; Yan, X.; Huang, B. Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes. *Ind. Eng. Chem. Res.* **2019**, *58*, 12899–12912. [[CrossRef](#)]
16. Jiang, B.; Xiang, J.; Wang, Y. Rolling bearing fault diagnosis approach using probabilistic principal component analysis denoising and cyclic bispectrum. *J. Vib. Control* **2016**, *22*, 2420–2433. [[CrossRef](#)]
17. Chen, H.; Jiang, B.; Lu, N.; Mao, Z. Deep PCA Based Real-time Incipient Fault Detection and Diagnosis Methodology for Electrical Drive in High-Speed Trains. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4819–4830. [[CrossRef](#)]
18. Wang, T.; Xu, H.; Han, J.; Elbouchikhi, E.; Benbouzid, M.E.H. Cascaded H-bridge multilevel inverter system fault diagnosis using a PCA and multiclass relevance vector machine approach. *IEEE Trans. Power Electron.* **2015**, *30*, 7006–7018. [[CrossRef](#)]
19. Sánchez-Fernández, A.; Fuente, M.J.; Sainz-Palmero, G.I. Fault detection in wastewater treatment plants using distributed PCA methods. In Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), Luxembourg, 8–11 September 2015; pp. 1–7. [[CrossRef](#)]
20. Kazemi, P.; Giralt, J.; Bengoa, C.; Masoumian, A.; Steyer, J.-P. Fault detection and diagnosis in water resource recovery facilities using incremental PCA. *Water Sci. Technol.* **2020**, *82*, 2711–2724. [[CrossRef](#)] [[PubMed](#)]
21. Cid-Galiot, J.J.; Aguilar-Lasserre, A.A.; Rodriguez-Jarquín, J.P.; Villanueva-Vasquez, D.; Posada-Gómez, R.; Juárez-Martínez, U. Fault-Tolerance Techniques and Global Positioning Systems Based on Artificial Intelligence for the Analysis of Corrosion Impact on the Pipelines of Oil and Gas Industry. *ASME J. Pressure Vessel Technol.* **2021**, *143*, 020801. [[CrossRef](#)]
22. Li, Z.; Tian, L.; Jiang, Q.; Yan, X. Dynamic nonlinear process monitoring based on dynamic correlation variable selection and kernel principal component regression. *J. Frankl. Inst.* **2022**, *359*, 4513–4539. [[CrossRef](#)]
23. Cheng, H.; Wu, J.; Liu, Y.; Huang, D. A novel fault identification and root-causality analysis of incipient faults with applications to wastewater treatment processes. *Chemom. Intell. Lab. Syst.* **2019**, *188*, 24–36. [[CrossRef](#)]
24. Bellamoli, F.; Di Iorio, M.; Vian, M.; Melgani, F. Machine learning methods for anomaly classification in wastewater treatment plants. *J. Environ. Manag.* **2023**, *344*, 118594. [[CrossRef](#)] [[PubMed](#)]
25. Khurshid, A.; Pani, A.K. Machine learning approaches for data-driven process monitoring of biological wastewater treatment plant: A review of research works on benchmark simulation model no. 1(bsm1). *Environ. Monit. Assess.* **2023**, *195*, 96. [[CrossRef](#)] [[PubMed](#)]
26. Sánchez-Fernández, A.; Baldan, F.J.; Sainz-Palmero, G.I.; Benitez, J.M.; Fuente, M.J. Fault detection based on time series modeling and multivariate statistical process control. *Chemom. Intell. Lab. Syst.* **2018**, *182*, 57–69. [[CrossRef](#)]
27. Newhart, K.B.; Holloway, R.W.; Hering, A.S.; Cath, T.Y. Data-driven performance analyses of wastewater treatment plants: A review. *Water Res.* **2019**, *157*, 498–513. [[CrossRef](#)]
28. Alvi, M.; Batstone, D.; Mbamba, C.K.; Keymer, P.; French, T.; Ward, A.; Dwyer, J.; Cardell-Oliver, R. Deep learning in wastewater treatment: A critical review. *Water Res.* **2023**, *245*, 120518. [[CrossRef](#)] [[PubMed](#)]
29. Ismail, W.; Niknejad, N.; Bahari, M.; Hendradi, R.; Zaizi, N.J.M.; Zulkifli, M.Z. Water treatment and artificial intelligence techniques: A systematic literature review research. *Environ. Sci. Pollut. Res.* **2023**, *30*, 71794–71812. [[CrossRef](#)] [[PubMed](#)]
30. Cicceri, G.; Maisano, R.; Morey, N.; Distefano, S. SWIMS: The Smart Wastewater Intelligent Management System. In Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP), Irvine, CA, USA, 23–27 August 2021; pp. 228–233. [[CrossRef](#)]
31. Liu, Y.; Ramin, P.; Flores-Alsina, X.; Gernaey, K.V. Transforming data into actionable knowledge for fault detection, diagnosis and prognosis in urban wastewater systems with AI techniques: A mini-review. *Process Saf. Environ. Prot.* **2023**, *172*, 501–512. [[CrossRef](#)]
32. Qiu, C.; Shao, G.; Zhang, Z.; Zhou, C.; Hou, Y.; Zhao, E.; Guo, X.; Guan, X. Unsupervised Real Time and Early Anomalies Detection Method for Sewer Networks Systems. *IEEE Access* **2024**, *12*, 21698–21709. [[CrossRef](#)]
33. Zhou, Q.; Situ, Z.; Teng, S.; Chen, G. Convolutional neural networks-based model for automated sewer defects detection and classification. *J. Water Resour. Plan. Manag.* **2021**, *147*, 04021036. [[CrossRef](#)]
34. Meseguer, J.; Puig, V.; Escobet, T. Fault Diagnosis Using a Timed Discrete-Event Approach Based on Interval Observers: Application to Sewer Networks. *IEEE Trans. Syst. Man Cybern.—Part A Syst. Hum.* **2010**, *40*, 900–916. [[CrossRef](#)]
35. Ocampo-Martinez, C.; Puig, V.; Quevedo, J.; Ingimundarson, A. Fault tolerant model predictive control applied on the Barcelona sewer network. In Proceedings of the 44th IEEE Conference on Decision and Control (CDC), Seville, Spain, 15 December 2005.
36. Meseguer, J.; Bernat, J.D.; Cembrano, G.; Puig, V. Fault-tolerant model predictive control applied to integrated urban drainage and sanitation systems for environmental protection. A: International Conference on Hydroinformatics. In Proceedings of the HIC 2018: 13th International Conference on Hydroinformatics, Palermo, Italia, 1–6 July 2018; Proceedings Book. pp. 1388–1396.

37. Saagi, R.; Flores-Alsina, X.; Fu, G.; Butler, D.; Gernaey, K.V.; Jeppsson, U. *Catchment & Sewer Network Simulation Model to Benchmark Control Strategies within Urban Wastewater Systems*; Elsevier: Amsterdam, The Netherlands, 2016.
38. Liu, J.; Wang, J.; Liu, X.; Ma, T.; Tang, Z. MWRSPCA: Online fault monitoring based on moving window recursive sparse principal component analysis. *J. Intell. Manuf.* **2022**, *33*, 1255–1271. [[CrossRef](#)]
39. Jiang, Q.; Gao, F.; Hui, Y.; Yan, X. Multivariate statistical monitoring of key operation units of batch processes based on time-slice CCA. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 1368–1375. [[CrossRef](#)]
40. Cembellín, A.; Francisco, M.; Vega, P. Distributed Model Predictive Control Applied to a Sewer System. *Processes* **2020**, *8*, 1595. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.