

Article

On Reduced Observer-Bank Synthesis and Edge Implementation for Residual Chlorine Concentration Soft Sensors in Water Distribution Networks

Nikolaos D. Kouvakas, Fotis N. Koumboulis *, Antonios N. Menexis, Dimitrios G. Fragakoulis and Maria P. Tzamtzi

Robotics, Automatic Control and Cyber-Physical Systems Laboratory, Department of Digital Industry Technologies, School of Science, National and Kapodistrian University of Athens, Euripus Campus, 34400 Euboea, Greece; nkouvak@dind.uoa.gr (N.D.K.); amenexis@dind.uoa.gr (A.N.M.); dfragakoulis@dind.uoa.gr (D.G.F.); mtzamtzi@dind.uoa.gr (M.P.T.)

* Correspondence: fkoumboulis@dind.uoa.gr

Abstract

In the present paper, a reduced observer-bank synthesis and edge-oriented implementation framework for a model-based residual chlorine concentration soft sensor in water distribution networks is presented. For a benchmark network described through nonlinear hydraulic and chlorine transport–reaction dynamics, local discrete-time linear approximants are derived around admissible operating points. Based on these approximants, a finite bank of full-order Luenberger-type observers is designed for the estimation of non-measurable chlorine concentration variables. The observer parameters are selected through a metaheuristic multicriterion tuning procedure that combines discrete-time pole-placement requirements with estimation-performance objectives. A central contribution of the paper is an observer-bank construction algorithm that eliminates redundant target operating areas. The algorithm adaptively covers the normalized input domain by generating each observer operating area after nonlinear steady-state computation, local linearization, observer tuning, and validation for each operating point. The resulting observer bank is combined with a switching supervisor based on target-area overlap and measurable-output convergence. The resulting scheme is implemented in a Node-RED edge runtime, where multiple observers execute in parallel and the supervisor/switching mechanism selects the active observer that provides the appropriate estimate. Deterministic telemetry generated by a MATLAB R2025b simulation of the water distribution network is fed to the Node-RED instance, enabling repeatable evaluation of acquisition, estimation, switching, single-stream telemetry arbitration, and runtime adaptation. Computational and edge runtime experiments demonstrate the feasibility of deploying the proposed switching-observer soft sensor for real-time chlorine monitoring in water distribution networks.

Academic Editors: Do Guen Yoo and Hyuk Jae Kwon

Received: 17 June 2026

Revised: 26 July 2026

Accepted: 3 August 2026

Published: 6 August 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: water distribution networks; residual chlorine soft sensors; discrete-time switching observers; observer-bank construction; observer-bank reduction; Node-RED edge implementation; simulation-supported edge validation

1. Introduction

Soft sensors are widely used when direct measurement of an important process variable is costly, delayed, intrusive, or technically difficult. They are commonly classified into data-driven and model-driven approaches, depending on whether the estimation mechanism is primarily learned from process data or derived from a physical/dynamic model (see [1,2] and the references therein). Observer-based soft sensors belong to the model-driven family. Instead of estimating the desired variable only through a static input–output relation, they propagate a dynamic state-space model and continuously correct the estimate through the measured-output residual. This is particularly useful when analytic process models can be produced, because convergence, sensitivity, and robustness can be addressed explicitly through Luenberger observers, Kalman-family filters, unknown-input observers, or related estimator structures (see [3–6] and the references therein).

Soft sensors have demonstrated broad applicability in systems where important variables are spatially distributed, inaccessible or costly to measure. Such applications include, among others, energy systems (indicatively see [7–10]), manufacturing and metallurgical processes (indicatively see [11–14]), water and water management applications (indicatively see [15–18]), and bioprocess and biomanufacturing applications (indicatively see [19–22]). Soft sensors constitute a generally applicable estimation framework for supplementing physical instrumentation and supporting monitoring, control, optimization, and fault-detection functions.

For nonlinear processes, a single linear observer is usually reliable only near the operating point for which it was designed. A practical way to retain the simplicity of linear observer design while covering a wider operating envelope is to construct a bank of local observers and switch among them according to the current operating region. This idea has been formalized in switching-observer soft sensor design, where local linear approximants, observer-oriented target operating areas, and rule-based switching are combined to estimate nonmeasurable process variables (see [2]). The approach was further extended through metaheuristic procedures for selecting observer parameters and organizing observer banks for soft sensor applications (see [23]). This line of work is directly connected with the present paper, since the proposed residual-chlorine soft sensor also relies on local linear approximants of nonlinear water-network dynamics, a bank of Luenberger-type observers, and a supervisory switching mechanism to maintain estimation accuracy during operating transitions (see [3,4]).

In water distribution networks, observer-based estimation has mainly been developed for hydraulic monitoring and diagnosis. Recent examples include extended Kalman filter (EKF) based hydraulic state estimation from sparse pressure and flow measurements (see [24]), nonlinear state observers for pipe-roughness calibration (see [25]), interval state-estimation methods (see [26]), unknown-input observer interpretations of EKF arrays for nodal-demand estimation (see [27]), and banks of Kalman observers for leak diagnosis (see [28]). These papers highlight the importance of model-based estimators in sparsely instrumented water distribution networks, but they mostly concern hydraulic states, demands, leakage, or calibration parameters. Relevant chlorine-oriented work includes data-driven chlorine soft sensing (see [29]), switching-observer chlorine soft sensors (see [3–5]), interval estimation of unmeasured chlorine states (see [30]), and ensemble Kalman filter data assimilation for chlorine state and reaction-parameter estimation (see [31]).

Soft sensor research has increasingly moved from purely simulation-based estimation toward deployable cyber–physical implementations. In wastewater treatment [32], a cost-effective Internet of Things (IoT) strategy for the remote deployment of a soft sensor is proposed. In [33], the authors present a full-scale study on the implementation of soft sensors for phosphorus and chemical oxygen demand estimation, supported by a non-intrusive industrial IoT infrastructure and a web-based visualization interface. Similar

implementation-oriented directions have also been reported for biochemical oxygen demand estimation using intelligence at the edge (see [34]), as well as for low-cost nitrate monitoring through an open-source soft sensor methodology that integrates IoT data streams and MQTT publish/subscribe messaging protocol based communication (see [35]). At the hardware level, embedded and on-device soft sensors have also received growing attention. Ling et al. proposed an on-device soft sensor framework for real-time fluid-flow estimation using microcontroller units or field-programmable gate array (FPGA) based execution, highlighting the advantages of moving inference from the cloud to local devices (see [36]). Related work has considered FPGA-oriented quantization strategies for real-time soft sensing (see [37]), automated construction of embedded FPGA soft sensors for wastewater-flow estimation (see [38]), and configurable multilayer perceptron based soft sensors on embedded FPGAs with different latency, precision, and resource-use targets (see [39]). Embedded soft sensor implementations have also been reported on Zynq system-on-chip platforms for bioprocess applications (see [40,41]), while earlier FPGA-based work addressed industrial soft sensing for kerosene freezing-point estimation (see [42]). Broader TinyML and edge-computing examples further confirm the feasibility of executing soft sensor models close to the physical process (see [43,44]). Nevertheless, most implementation-oriented studies are data-driven or application-specific, whereas edge implementation of a physics-based, switching-observer residual-chlorine soft sensor remains much less explored.

The present work contributes to this direction by implementing a bank of discrete-time observers in Node-RED, acquiring measurements using MQTT for cloud-supervised switching, and validating the complete estimation pipeline under simulation-supported edge runtime conditions. In particular, it develops an edge-oriented implementation framework for a residual chlorine concentration soft sensor in a benchmark water distribution network. The proposed soft sensor is based on a finite bank of discrete-time Luenberger-type observers, each designed around a nominal operating point of the nonlinear chlorine-transport model. For every nominal point, the corresponding steady-state model is computed, and a local discrete-time linear approximant is constructed and shown to be both reconstructable and observable. The observer parameters are tuned through a metaheuristic/multicriterion optimization procedure that combines pole-placement requirements with estimation-performance objectives. A central contribution of the paper is the observer-bank construction algorithm, which systematically builds the observer bank as an adaptive covering of the normalized input domain, reducing the total number of observers included in the bank. The resulting bank and its switching supervisor are then transferred to a Node-RED edge runtime, where the observers execute in parallel and the supervisor selects the active estimate. Deterministic telemetry generated by a MATLAB R2025b simulation of the water distribution network is fed to the Node-RED instance, enabling reproducible validation of the runtime behavior, single-stream telemetry arbitration, and continuous adaptation of the implemented soft sensor during operating transitions. The proposed implementation is shown to provide highly accurate estimates successfully reconstructing all nonmeasurable concentrations.

2. Mathematical Modeling of a Benchmark Water Distribution Network

Adequate free chlorine residual must be maintained throughout a water distribution network, as it constitutes a barrier against microbial regrowth and post-treatment contamination. However, chlorine concentration is inherently nonuniform over the network and evolves under the combined effects of hydraulic transients, demand variability, water age, and bulk and wall reactions. Since continuous in-line chlorine measurement at all critical locations is not practically feasible, soft sensors are required to estimate chlorine

concentration at uninstrumented points from routinely available telemetry sampled in discrete time. In the present work, the soft sensor is derived using a physics-based approach. This requires a mathematical model of the network that captures the coupling between hydraulic behavior and chlorine transport and decay, supports observer design with analysable convergence properties, and reduces dependence on purely black-box predictors, whose performance may deteriorate under changing operating conditions. In the present paper, the benchmark network shown in Figure 1 is used. The network has a loop-free, tree-type topology and consists of twelve hydraulic nodes interconnected through eleven conduits. Five of the nodes, namely Nodes 2, 4, 5, 6, and 10, represent internal junctions at which aggregated water demands are imposed. The remaining nodes, namely Nodes 1, 3, 7, 8, 9, 11, and 12, are terminal boundary-head nodes at which the corresponding hydraulic boundary conditions are imposed. The network contains several successive branching levels. Conduit 1 connects boundary Node 1 to the first demand junction, Node 2. From Node 2, the network is divided into two branches. Conduit 2 leads to boundary Node 3, whereas Conduit 3 leads to the second demand junction, Node 4. The network is further divided at Node 4, with Conduits 4 and 5 connecting it to demand Nodes 5 and 6, respectively. Node 5 is connected to the terminal boundary Nodes 7 and 8 through Conduits 6 and 7. Similarly, Node 6 is connected to boundary Node 9 through Conduit 8 and to demand Node 10 through Conduit 9. The final branching level occurs at Node 10, which is connected to the terminal boundary Nodes 11 and 12 through Conduits 10 and 11, respectively. The selected topology introduces spatially distributed water withdrawals, successive hydraulic divisions, and transport paths of different lengths. Consequently, hydraulic variations occurring at an upstream junction propagate through several downstream branches, while the chlorine concentration in each conduit depends on the local flow conditions, the upstream concentration, and the accumulated transport and reaction effects along the corresponding path.

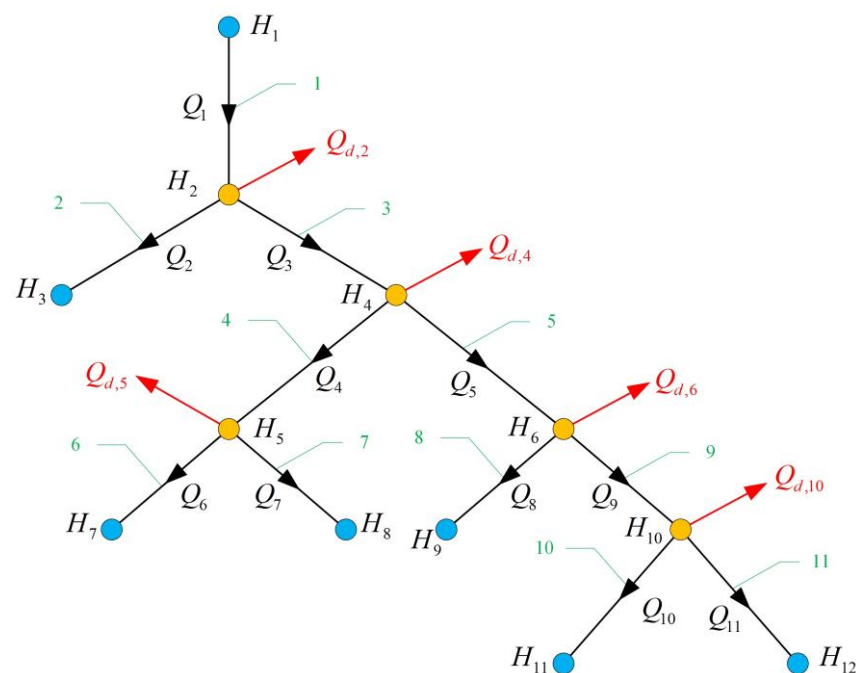


Figure 1. Benchmark water distribution network.

2.1. Continuous Time Nonlinear Model

The mathematical model describing flow and chlorine concentration dynamics can be approximated by a nonlinear state-space model of the form

$$\mathbf{E}\dot{\mathbf{x}} = \mathbf{\Gamma}(\mathbf{x}) + \mathbf{Z}(\mathbf{x})\mathbf{u}, \quad (1)$$

where

$$\begin{aligned} \mathbf{x} &= [x_1 \ \cdots \ x_{38}]^T = [Q_1 \ Q_2 \ Q_3 \ Q_4 \ Q_5 \ Q_6 \ Q_7 \ Q_8 \ Q_9 \ Q_{10} \ Q_{11} \ \vdots \\ &\vdots \ H_2 \ H_4 \ H_5 \ H_6 \ H_{10} \ c_{1,1} \ c_{1,2} \ c_{2,1} \ c_{2,2} \ c_{3,1} \ c_{3,2} \ c_{4,1} \ c_{4,2} \ c_{5,1} \ c_{5,2} \ \vdots \\ &\vdots \ c_{6,1} \ c_{6,2} \ c_{7,1} \ c_{7,2} \ c_{8,1} \ c_{8,2} \ c_{9,1} \ c_{9,2} \ c_{10,1} \ c_{10,2} \ c_{11,1} \ c_{11,2}]^T \\ \mathbf{u} &= [u_1 \ \cdots \ u_{13}]^T = [H_1 \ H_3 \ H_7 \ H_8 \ H_9 \ H_{11} \ H_{12} \ c_0 \ \vdots \\ &\vdots \ Q_{d,2} \ Q_{d,4} \ Q_{d,5} \ Q_{d,6} \ Q_{d,10}]^T \\ \mathbf{\Gamma}(\mathbf{x}) &= [\gamma_j(\mathbf{x})] \in \mathbb{R}^{38 \times 1}, \quad \mathbf{Z}(\mathbf{x}) = [z_{i,j}(\mathbf{x})] \in \mathbb{R}^{38 \times 13}. \end{aligned}$$

Their nonzero elements are presented in Appendix A. Note that H_1 , H_3 , H_7 , H_8 , H_9 , H_{11} and H_{12} are boundary (actuatable) pressure heads imposed by reservoirs at the corresponding nodes; Q_i is the volumetric flow rate in conduit i , $Q_{d,2}$, $Q_{d,4}$, $Q_{d,5}$, $Q_{d,6}$ and $Q_{d,10}$ are aggregated consumption flow rates at internal Nodes 2, 4, 5, 6 and 10, respectively; $c_{i,j}$ is the chlorine concentration at section j of conduit i ; c_0 is the influent chlorine concentration to the network from Node 1. The matrix $\mathbf{E}$ is equal to $\mathbf{I}_{38}$ where $\mathbf{I}_\alpha$ is the $\alpha \times \alpha$ identity matrix. If the hydraulic subsystem of the model is assumed to be static, a quasi-static flow-chlorine model is resulted (see [4]); it takes on the block diagonal form $\mathbf{E} = \text{diag}\{\mathbf{0}_{16 \times 16}, \mathbf{I}_{22}\}$, where $\mathbf{0}_{\alpha \times \beta}$ is the $\alpha \times \beta$ zero matrix.

2.2. Continuous and Discrete-Time Linear Approximant

Let $\bar{u}_j$ ($j=1, \dots, 13$) be the nominal values of the inputs of the system and $\bar{x}_i$ ($i=1, \dots, 38$) be the corresponding nominal values of the state variables. Furthermore, let $\bar{\mathbf{u}} = [\bar{u}_1 \ \cdots \ \bar{u}_{13}]^T$ and $\bar{\mathbf{x}} = [\bar{x}_1 \ \cdots \ \bar{x}_{38}]^T$ be the vectors of nominal values of the inputs and the state variables. The operating vector of the system's dynamics is denoted by the pair $\bar{\mathbf{o}} = (\bar{\mathbf{u}}, \bar{\mathbf{x}})$, where its elements satisfy the following equality

$$\mathbf{\Theta}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \mathbf{\Gamma}(\bar{\mathbf{x}}) + \mathbf{Z}(\bar{\mathbf{x}})\bar{\mathbf{u}} = \mathbf{0}_{38 \times 1}. \quad (2)$$

In the case of constant flow directions and applying a series of computations, it can be verified that relation (2) is satisfied if and only if the equalities in Appendix B are satisfied. Applying a series of computations, it can be verified that the linear approximant of the nonlinear model (1) is of the form

$$\mathbf{E}\delta\dot{\mathbf{x}} = \mathbf{J}_s(\bar{\mathbf{x}}, \bar{\mathbf{u}})\delta\mathbf{x} + \mathbf{Z}(\bar{\mathbf{x}})\delta\mathbf{u}, \quad (3)$$

where $\delta\mathbf{x}$ is the response of the linear approximant (3) for $\delta\mathbf{u} = \Delta\mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$ that approximates $\Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$ around the operating point $\bar{\mathbf{o}} = (\bar{\mathbf{u}}, \bar{\mathbf{x}})$, and where $\mathbf{J}_s(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \partial\mathbf{\Theta}(\mathbf{x}, \mathbf{u})/\partial\mathbf{x}|_{\mathbf{u}=\bar{\mathbf{u}}, \mathbf{x}=\bar{\mathbf{x}}}$. It is important to mention that the nonlinear algebraic system of equations in (2) is solvable with respect to $\bar{\mathbf{x}}$ (see [3,4]). Hence, there exists a nonlinear vector function that maps the nominal values of the inputs to the nominal values of the states, i.e., $\bar{\mathbf{x}} = \boldsymbol{\sigma}(\bar{\mathbf{u}})$. Hence, the linear approximant system matrices can be expressed as $\mathbf{A}(\bar{\mathbf{u}}) = \mathbf{J}_s(\boldsymbol{\sigma}(\bar{\mathbf{u}}), \bar{\mathbf{u}})$, $\mathbf{B}(\bar{\mathbf{u}}) = \mathbf{Z}(\boldsymbol{\sigma}(\bar{\mathbf{u}}))$ and the linear approximant becomes

$$\mathbf{E}\delta\dot{\mathbf{x}} = \mathbf{A}(\bar{\mathbf{u}})\delta\mathbf{x} + \mathbf{B}(\bar{\mathbf{u}})\delta\mathbf{u}. \quad (4)$$

Define

$$\begin{aligned} \delta\mathbf{z} &= [\delta z_1 \quad \cdots \quad \delta z_{16}]^T = [\delta x_1 \quad \cdots \quad \delta x_{16}]^T, \\ \delta\mathbf{q} &= [\delta q_1 \quad \cdots \quad \delta q_{22}]^T = [\delta x_{17} \quad \cdots \quad \delta x_{38}]^T. \end{aligned}$$

Applying a series of computations to (4), it can be observed that the linear approximant can be rewritten as

$$\mathbf{A}_{1,1}(\bar{\mathbf{u}})\delta\mathbf{z} + \mathbf{B}_1(\bar{\mathbf{u}})\delta\mathbf{u} = \mathbf{0}_{16 \times 1}, \quad (5)$$

$$\delta\dot{\mathbf{q}} = \mathbf{A}_{2,1}(\bar{\mathbf{u}})\delta\mathbf{z} + \mathbf{A}_{2,2}(\bar{\mathbf{u}})\delta\mathbf{q} + \mathbf{B}_2(\bar{\mathbf{u}})\delta\mathbf{u}. \quad (6)$$

In zero-order-hold discretization conditions, i.e., if the input vector $\delta u(t)$ is piecewise constant over each sampling interval, the exact discrete-time model at sampling instants $t = kT_s$, where $k \in \mathbb{N}_0$ and $T_s \in \mathbb{R}^+$, takes on the form

$$\delta\mathbf{q}(k+1) = \mathbf{A}_d(\bar{\mathbf{u}})\delta\mathbf{q}(k) + \mathbf{B}_d(\bar{\mathbf{u}})\delta\mathbf{u}(k), \quad (7)$$

where

$$\begin{aligned} \mathbf{A}_d(\bar{\mathbf{u}}) &= \left[(a_d)_{i,j} \right] = \exp(\mathbf{A}_{2,2}(\bar{\mathbf{u}})T_s), \\ \mathbf{B}_d(\bar{\mathbf{u}}) &= \left[(b_d)_{i,j} \right] = \int_0^{T_s} \exp(\mathbf{A}_{2,2}(\bar{\mathbf{u}})\tau) \left(\mathbf{B}_2(\bar{\mathbf{u}}) - \mathbf{A}_{2,1}(\bar{\mathbf{u}})\mathbf{A}_{1,1}(\bar{\mathbf{u}})^{-1}\mathbf{B}_1(\bar{\mathbf{u}}) \right) d\tau. \end{aligned}$$

It is important to mention that the discrete-time linear approximant in (7) is reconstructable (see [2]). The measurable variables are only the chlorine concentrations at the exit of Conduits 2, 6, 7, 8, 10 and 11, i.e., δq_4 , δq_{12} , δq_{14} , δq_{16} , δq_{20} and δq_{22} . Thus, the measurable output vector is $\delta\mathbf{y}(k) = \mathbf{C}_d\delta\mathbf{q}(k)$, where the nonzero elements of $\mathbf{C}_d = \left[(c_d)_{i,j} \right] \in \mathbb{R}^{6 \times 22}$ are $(c_d)_{1,4} = (c_d)_{2,12} = (c_d)_{3,14} = (c_d)_{4,16} = (c_d)_{5,20} = (c_d)_{6,22} = 1$.

3. Soft Sensor Design

3.1. Soft Sensor Structure

In the present section, based on the discrete-time linear approximant (7), a soft sensor for the estimation of chlorine concentration inside the network will be designed. The soft sensor will be based on discrete-time full-order Luenberger type observers of the form

$$\mathfrak{Z}:\delta\hat{\mathbf{q}}(k+1) = (\mathbf{A}_d(\bar{\mathbf{u}}) - \mathbf{L}(\bar{\mathbf{u}})\mathbf{C}_d)\delta\hat{\mathbf{q}}(k) + \mathbf{B}_d(\bar{\mathbf{u}})\delta\mathbf{u}(k) + \mathbf{L}(\bar{\mathbf{u}})\delta\mathbf{y}(k); \quad \delta\hat{\mathbf{q}}(0) = \delta\hat{\mathbf{q}}_0 \quad (8)$$

where $\delta\hat{\mathbf{q}}$ is the estimate of $\delta\mathbf{q}$ and $\mathbf{L}(\bar{\mathbf{u}}) \in \mathbb{R}^{22 \times 6}$ is an appropriate observer matrix, to be selected by the designer. For pre-selected operating points, different observers will be designed and an appropriate switching mechanism between them will be implemented. Applying elementary computations, it can be verified that the linear approximant in (7) is observable. Hence, the elements of $\mathbf{L}$ can be selected by the designer such that the eigenvalues of $\mathbf{F}(\bar{\mathbf{u}}) = \mathbf{A}_d(\bar{\mathbf{u}}) - \mathbf{L}(\bar{\mathbf{u}})\mathbf{C}_d$ are appropriately adjusted. It can readily be verified that the discrete-time linear approximant (7) is observable, hence, arbitrary eigenvalue assignment of $\mathbf{F}(\bar{\mathbf{u}})$ can be achieved by appropriately selecting the elements of $\mathbf{L}$. Let

$$p_0(z) = \det(z\mathbf{I}_{22} - \mathbf{F}(\bar{\mathbf{u}})), \quad (9)$$

$$p_s(z) = \det(z\mathbf{I}_{22} - \mathbf{A}_d(\bar{\mathbf{u}})). \quad (10)$$

Let $\pi_{o,i}$ ($i=1,\dots,22$) be the roots of $p_o(z)$ and $\pi_{s,i}$ ($i=1,\dots,22$) be the roots of $p_s(z)$. Without loss of generality, assume that $|\pi_{o,i}| \leq |\pi_{o,i+1}|$ and $|\pi_{s,i}| \leq |\pi_{s,i+1}|$ ($i=1,\dots,21$). In what follows, the observer polynomial roots are restricted to satisfy the following constraints:

- The roots of $p_o(z)$ are real, nonnegative and stable, i.e., $\pi_{o,i} \in [0,1)$,
- The roots of $p_o(z)$ are ordered and separated by minimum distance being equal to $\gamma \in \mathbb{R}^+$, i.e., it holds that $\pi_{o,i+1} > \gamma + \pi_{o,i}$,
- Regional stability is achieved, i.e., it holds that $\pi_{o,i} < |\pi_{s,i}|$,
- The H_2 norm of the resolvent matrix $\mathbf{R}(z) = (z\mathbf{I}_{22} - \mathbf{F}(\bar{\mathbf{u}}))^{-1}$ of the error dynamics is minimized.

Regarding the last design goal, minimization of the H_2 norm of the resolvent matrix of the error dynamics corresponds to minimizing the root-mean-square energy gain from the initial estimation error to the error transient. Applying a series of computations to (9), it can be verified that the design requirement of pole placement can be achieved by selecting the elements of the first column of $\mathbf{L}$. Once the poles are fixed, the elements of the second column of $\mathbf{L}$ can be used to satisfy the norm-based design requirement. The above design requirements lead to a multicriteria optimization problem which is generally nonlinear and nonconvex. For this reason, a numerical optimization approach is required.

3.2. Observer-Parameter Tuning Algorithm

Given the complexity involved in deriving the observer degrees of freedom presented in the previous subsection, here, toward numerical minimization of the H_2 norm of the resolvent matrix of the error dynamics, an eigenstructure-based approach for the derivation of the observer-gain elements (see [45–48]) will be used, in combination with a metaheuristic optimization algorithm, similar to that presented in [3,49,50].

For a fixed operating point $\bar{\mathbf{u}}$, let $\mathbf{v}_i \in \mathbb{R}^{22 \times 1}$ ($i=1,\dots,22$) be the corresponding left-eigenvector directions of the error-dynamics matrix corresponding to the observer pole $\pi_{o,i}$. The required eigenstructure therefore satisfies

$$\mathbf{F}^T(\bar{\mathbf{u}})\mathbf{v}_i(\bar{\mathbf{u}}) = \pi_{o,i}(\bar{\mathbf{u}})\mathbf{v}_i(\bar{\mathbf{u}}) \quad (11)$$

Let $\mathbf{z}_i(\bar{\mathbf{u}}) \in \mathbb{R}^{6 \times 1}$ be the output-direction vector, determined through

$$\mathbf{z}_i(\bar{\mathbf{u}}) = \mathbf{L}^T(\bar{\mathbf{u}})\mathbf{v}_i(\bar{\mathbf{u}}) \quad (12)$$

Substituting (12) into (11) and applying a series of computations results in

$$(\mathbf{A}_d^T(\bar{\mathbf{u}}) - \pi_{o,i}(\bar{\mathbf{u}})\mathbf{I}_{22})\mathbf{v}_i(\bar{\mathbf{u}}) = \mathbf{C}_d^T\mathbf{z}_i(\bar{\mathbf{u}}) \quad (13)$$

Provided that $\mathbf{A}_d^T(\bar{\mathbf{u}}) - \pi_{o,i}(\bar{\mathbf{u}})\mathbf{I}_{22}$ is nonsingular, the left-eigenvector direction associated with the i -th prescribed pole is determined through

$$\mathbf{v}_i(\bar{\mathbf{u}}) = (\mathbf{A}_d^T(\bar{\mathbf{u}}) - \pi_{o,i}(\bar{\mathbf{u}})\mathbf{I}_{22})^{-1}\mathbf{C}_d^T\mathbf{z}_i(\bar{\mathbf{u}}). \quad (14)$$

The pair $(\mathbf{v}_i(\bar{\mathbf{u}}), \mathbf{z}_i(\bar{\mathbf{u}}))$ is invariant with respect to multiplication by the same non-zero scalar. Hence, one component of every output-direction vector may be fixed without restricting the attainable observer matrices. Let $a_i \in \{1,\dots,6\}$ denote the anchor index associated with the i -th pole, let $\mathbf{e}_{a_i}(\bar{\mathbf{u}}) \in \mathbb{R}^6$ be the corresponding canonical vector, and

let $\mathbf{E}_{a_i} \in \mathbb{R}^{6 \times 5}$ be the matrix obtained from $\mathbf{I}_6$ after deleting its a_i -th column. The output direction is parameterized as

$$\mathbf{z}_i(\bar{\mathbf{u}}) = \mathbf{e}_{a_i}(\bar{\mathbf{u}}) + \mathbf{E}_{a_i} \mathbf{d}_i(\bar{\mathbf{u}}); \quad \mathbf{d}_i(\bar{\mathbf{u}}) \in \mathbb{R}^{5 \times 1} \quad (15)$$

Thus, the anchor component of $\mathbf{z}_i(\bar{\mathbf{u}})$ is equal to one, while its remaining five components constitute free observer parameters. To avoid normalization with respect to a component of very small magnitude, the anchor index is selected from an initial feasible observer as $a_i = \operatorname{argmax}_{r \in \{1, \dots, 6\}} |z_{r,i}^{(0)}|$ and is retained unchanged during the optimization. Define the modal matrix, the output-direction matrix and the diagonal matrix of prescribed observer poles as

$$\mathbf{V}(\bar{\mathbf{u}}) = [\mathbf{v}_1(\bar{\mathbf{u}}) \quad \dots \quad \mathbf{v}_{22}(\bar{\mathbf{u}})] \quad (16)$$

$$\mathbf{Z}(\bar{\mathbf{u}}) = [\mathbf{z}_1(\bar{\mathbf{u}}) \quad \dots \quad \mathbf{z}_{22}(\bar{\mathbf{u}})] \quad (17)$$

$$\mathbf{\Pi}_o(\bar{\mathbf{u}}) = \operatorname{diag}(\pi_{o,1}(\bar{\mathbf{u}}), \dots, \pi_{o,22}(\bar{\mathbf{u}})). \quad (18)$$

If $\mathbf{V}(\bar{\mathbf{u}})$ is nonsingular, the relations $\mathbf{L}^T(\bar{\mathbf{u}}) \mathbf{v}_i(\bar{\mathbf{u}}) = \mathbf{z}_i(\bar{\mathbf{u}})$, $i = 1, \dots, 22$ uniquely determine the complete observer matrix. Clearly, it holds that

$$\mathbf{L}^T(\bar{\mathbf{u}}) \mathbf{V}(\bar{\mathbf{u}}) = \mathbf{Z}(\bar{\mathbf{u}}) \quad (19)$$

$$\mathbf{L}(\bar{\mathbf{u}}) = \left(\mathbf{Z}(\bar{\mathbf{u}}) \mathbf{V}(\bar{\mathbf{u}})^{-1} \right)^T \quad (20)$$

Using the definitions of $\mathbf{v}_i(\bar{\mathbf{u}})$ and $\mathbf{z}_i(\bar{\mathbf{u}})$ gives

$$\mathbf{F}^T(\bar{\mathbf{u}}) \mathbf{V}(\bar{\mathbf{u}}) = \left(\mathbf{A}_d^T(\bar{\mathbf{u}}) - \mathbf{C}_d^T \mathbf{L}(\bar{\mathbf{u}})^T \right) \mathbf{V}(\bar{\mathbf{u}}) \quad (21)$$

or equivalently

$$\mathbf{F}^T(\bar{\mathbf{u}}) \mathbf{V}(\bar{\mathbf{u}}) = \mathbf{V}(\bar{\mathbf{u}}) \mathbf{\Pi}_o(\bar{\mathbf{u}}) \quad (22)$$

and therefore

$$\mathbf{F}^T(\bar{\mathbf{u}}) = \mathbf{V}(\bar{\mathbf{u}}) \mathbf{\Pi}_o(\bar{\mathbf{u}}) \mathbf{V}^{-1}(\bar{\mathbf{u}}) \quad (23)$$

Consequently, the eigenvalues of $\mathbf{A}_d(\bar{\mathbf{u}}) - \mathbf{L}(\bar{\mathbf{u}}) \mathbf{C}_d$ are precisely $\pi_{o,i}$ ($i = 1, \dots, 22$). For improved numerical scaling, the columns of $\mathbf{V}$ and $\mathbf{Z}$ may be normalized by the Euclidean norms of the corresponding vectors $\mathbf{v}_i$. Defining

$$\mathbf{S}(\bar{\mathbf{u}}) = \operatorname{diag}(\|\mathbf{v}_1(\bar{\mathbf{u}})\|_2, \dots, \|\mathbf{v}_{22}(\bar{\mathbf{u}})\|_2)$$

defines $\tilde{\mathbf{V}}(\bar{\mathbf{u}}) = \mathbf{V}(\bar{\mathbf{u}}) \mathbf{S}^{-1}(\bar{\mathbf{u}})$ and $\tilde{\mathbf{Z}}(\bar{\mathbf{u}}) = \mathbf{Z}(\bar{\mathbf{u}}) \mathbf{S}^{-1}(\bar{\mathbf{u}})$. Clearly, it holds that $\tilde{\mathbf{Z}}(\bar{\mathbf{u}}) \tilde{\mathbf{V}}^{-1}(\bar{\mathbf{u}}) = \mathbf{Z}(\bar{\mathbf{u}}) \mathbf{V}^{-1}(\bar{\mathbf{u}})$. Hence, the common column normalization does not alter the resulting observer matrix. It only improves the numerical scaling of its reconstruction.

Accordingly, the observer degrees of freedom used by the metaheuristic optimization algorithm are the 22 prescribed observer poles and the five free output-direction components associated with each pole. The optimization vector is

$$\boldsymbol{\chi}(\bar{\mathbf{u}}) = [\pi_{o,1}(\bar{\mathbf{u}}) \quad \dots \quad \pi_{o,22}(\bar{\mathbf{u}}) \quad \mathbf{d}_1^T(\bar{\mathbf{u}}) \quad \dots \quad \mathbf{d}_{22}^T(\bar{\mathbf{u}})]^T \in \mathbb{R}^{132 \times 1} \quad (24)$$

Clearly, the H_2 norm of the resolvent matrix of the dynamics of the error is a function of $\boldsymbol{\chi}$ and $\bar{\mathbf{u}}$, i.e., it holds that

$$f(\bar{\mathbf{u}}) = \|\mathbf{R}(z)\|_2. \quad (25)$$

Note that $\|\mathbf{R}(z)\|_2^2$ can be determined through

$$\|\mathbf{R}(z)\|_2^2 = \text{tr}(\mathbf{P}(\bar{\mathbf{u}})), \quad (26)$$

where $\mathbf{P}(\bar{\mathbf{u}})$ is the unique positive semidefinite solution of the discrete Lyapunov equation

$$\mathbf{P}(\bar{\mathbf{u}}) = (\mathbf{A}_d(\bar{\mathbf{u}}) - \mathbf{L}(\bar{\mathbf{u}})\mathbf{C}_d)\mathbf{P}(\bar{\mathbf{u}})(\mathbf{A}_d(\bar{\mathbf{u}}) - \mathbf{L}(\bar{\mathbf{u}})\mathbf{C}_d)^T + \mathbf{I}_6. \quad (27)$$

Since the system and observer matrices depend upon the operating point, minimization of $f(\bar{\mathbf{u}})$ will be performed for each operating point $\bar{\mathbf{u}}$.

The proposed optimization algorithm is based on a constrained, iterative random-search metaheuristic approach defined over a progressively refined hyperrectangular region of the observer-parameter space. At each stage, valid parameter sets are randomly generated within the search box. The sampling process is organized into multiple repetitions, each producing a local feasible solution, while the best solution over the entire optimization run is retained separately. Upon completion of all repetitions, the search region is updated according to the dispersion of the best solutions generated during the repetitions, thereby shrinking the parameter search area. The procedure is repeated until either the maximum number of objective-function evaluations is reached or the normalized half-widths of the search box become smaller than the prescribed tolerance. The constrained metaheuristic algorithm was selected because of its transparent operation and straightforward implementation. Every candidate observer is synthesized according to explicit upper/lower bounds on the design parameters, validated against the stability and gain constraints, and analyzed regarding a given performance specification, thereby providing an intuitive and reproducible observer design procedure. Additionally, the algorithm can easily be extended to accommodate higher model dimensions and more design constraints. Because it runs offline, it does not affect the real-time performance of the soft sensor. Let

- J_{best} be the optimal objective value found over the whole run,
- χ^* be the corresponding best observer-parameter vector,
- $(\chi_j)_c^{(m)}$ be the center of the search interval for parameter j at refinement step m ,
- $(\chi_j)_w^{(m)}$ be the half-width of the search interval for parameter j at refinement step m ,
- n_{rep} be the number of repetitions per refinement step,
- n_{loop} be the number of evaluations per repetition,
- n_{total} be the maximum total number of objective-function evaluations,
- n_{feas} be the maximum number of resampling attempts to find a feasible point in the search box,
- ε be the termination tolerance.

The metaheuristic optimization algorithm is summarized in Algorithm 1:

Algorithm 1. Metaheuristic optimization algorithm.

-
- Set $N_{eval} = 0$ and $J_{best} = \infty$. Choose the initial search-box center and half-
- S0:** widths $(\chi_j)_c^{(0)}$ and $(\chi_j)_w^{(0)}$ ($j=1, \dots, 132$). Set the refinement index $m = 0$.
Define the current search space
- S1:** $\mathcal{D}^{(m)} = \left\{ \chi \in \mathbb{R}^{12 \times 1} : (\chi_j)_c^{(m)} - (\chi_j)_w^{(m)} \leq \chi_j \leq (\chi_j)_c^{(m)} + (\chi_j)_w^{(m)}, j=1, \dots, 132 \right\}$
- S2:** Set $i_1 = 0$.
- S3:** Set $i_1 = i_1 + 1$ and $i_2 = 0$.
- S4:** Set the feasibility-resampling counter $r = 0$.
Randomly generate a candidate vector
- S5:** $\chi^{(i_1, i_2+1)} = \left[(\chi_1)^{(i_1, i_2+1)} \quad \dots \quad (\chi_{132})^{(i_1, i_2+1)} \right]^T \in \mathcal{D}^{(m)}$ and set $r = r + 1$.
Check if the constraints presented in Section 3.1 are satisfied. If they are not
- S6:** satisfied and $r < n_{feas}$, go to S5. If they are not satisfied and $r = n_{feas}$, terminate the algorithm unsuccessfully. If they are satisfied, continue.
- S7:** Set $i_2 = i_2 + 1$ and $N_{eval} = N_{eval} + 1$. Evaluate $J_{i_1, i_2} = f(\bar{\mathbf{u}}, \chi^{(i_1, i_2)})$.
- S8:** If $J_{i_1, i_2} < J_{best}$ then set $J_{best} = J_{i_1, i_2}$ and $\chi^* = \chi^{(i_1, i_2)}$.
- S9:** If $N_{eval} = n_{total}$ then go to S15.
- S10:** If $i_2 < n_{loop}$ then go to S4.
Find $J_{i_1, \min} = \min \{ J_{i_1, i_2}, i_2 = 1, \dots, n_{loop} \}$ and the corresponding observer-parameter vector $\chi_{\min}^{(i_1)}$.
- S11:** If $i_1 < n_{rep}$ then go to S3.
Determine the observer parameter vectors $\chi_{\min}$ and $\chi_{\max}$ corresponding to
- S13:** $J_{\min} = \min \{ J_{i_1, \min}, i_1 = 1, \dots, n_{rep} \}$ and $J_{\max} = \max \{ J_{i_1, \min}, i_1 = 1, \dots, n_{rep} \}$, respectively.
Update the search-box center and half-widths according to
- $$(\chi_j)_c^{(m+1)} = \frac{(\chi_{\min})_j + (\chi_{\max})_j}{2}, \quad (\chi_j)_w^{(m+1)} = \frac{|(\chi_{\min})_j - (\chi_{\max})_j|}{2} \quad (j=1, \dots, 132)$$
- S14:** If
- $$\max_{j=1, \dots, 12} \frac{(\chi_j)_w^{(m+1)}}{\max \left\{ 1, |(\chi_j)_c^{(m+1)}| \right\}} > \varepsilon$$
- then set $m = m + 1$ and go to S1. Otherwise, go to S15.
- S15:** Return the best observer-parameter vector found $\chi = \chi^*$.
-

3.3. Target Operating Areas Derivation

As mentioned in the previous subsections, each observer in the soft sensor bank is designed for a specific operating point. Considering that the observers are linear and the process is inherently nonlinear, it is imperative to determine the operating area of each observer, namely, the input region in which it behaves satisfactorily. For a given observer, designed around an operating point, let $\bar{\mathbf{u}}$, its operating area, is defined as a 13-dimensional hypercube of the form

$$\mathbb{H}(\bar{\mathbf{u}}) = \left\{ \tilde{\mathbf{u}} \in \mathbb{R}^{13 \times 1} : u_i^- (\bar{\mathbf{u}}) \leq \tilde{u}_i \leq u_i^+ (\bar{\mathbf{u}}), i=1, \dots, 13 \right\}, \quad (28)$$

where $\tilde{u}_i$ denotes the steady-state value of u_i ($i=1, \dots, 13$) during a stepwise transition, $\tilde{\mathbf{u}}$ is the corresponding steady-state vector, while u_i^- and u_i^+ are the lower and upper

bounds of the corresponding input variables. The 13-dimensional hypercube in (28) can be rewritten in an equivalent Cartesian product form as

$$\mathbb{H}(\bar{\mathbf{u}}) = \times_{i=1}^{13} [u_i^-(\bar{\mathbf{u}}), u_i^+(\bar{\mathbf{u}})], \quad (29)$$

where $\times_{i=\alpha}^{\beta} \mathbb{Q}_i$ denotes the Cartesian product of the argument ranges $\mathbb{Q}_i$ ($i = \alpha, \dots, \beta$), i.e., $\times_{i=\alpha}^{\beta} \mathbb{Q}_i = \mathbb{Q}_{\alpha} \times \dots \times \mathbb{Q}_{\beta}$. Define $\tilde{\boldsymbol{\zeta}} = [\tilde{\mathbf{q}}^T \ \tilde{\mathbf{u}}^T]^T$, $\bar{\boldsymbol{\zeta}} = [\bar{\mathbf{q}}^T \ \bar{\mathbf{u}}^T]^T$ and $\tilde{\boldsymbol{\zeta}}_e = [\tilde{\mathbf{q}}_e^T \ \tilde{\mathbf{u}}^T]^T$, where $\tilde{\mathbf{q}} = [\tilde{q}_1 \ \dots \ \tilde{q}_{22}]^T = [\tilde{x}_{17} \ \dots \ \tilde{x}_{38}]^T$ is the steady-state vector of the chlorine concentration variables after application of the stepwise transition, $\tilde{\mathbf{q}}_e = [\tilde{q}_{e,1} \ \dots \ \tilde{q}_{e,22}]^T = [\tilde{x}_{e,17} \ \dots \ \tilde{x}_{e,38}]^T$ is the steady-state estimation of the chlorine concentration variables determined through the observer in (8) and where $\bar{\mathbf{q}} = [\bar{q}_1 \ \dots \ \bar{q}_{22}]^T = [\bar{x}_{17} \ \dots \ \bar{x}_{38}]^T$ is the corresponding vector of nominal values. The observer's steady-state performance is quantified by the normalized steady-state estimation error

$$\varepsilon_{ss} = \sqrt{\frac{[\mathbf{W}(\tilde{\boldsymbol{\zeta}} - \tilde{\boldsymbol{\zeta}}_e)]^T [\mathbf{W}(\tilde{\boldsymbol{\zeta}} - \tilde{\boldsymbol{\zeta}}_e)]}{[\mathbf{W}(\tilde{\boldsymbol{\zeta}} - \bar{\boldsymbol{\zeta}})]^T [\mathbf{W}(\tilde{\boldsymbol{\zeta}} - \bar{\boldsymbol{\zeta}})]}} \times 100\%, \quad (30)$$

where

$$\mathbf{W} = \begin{bmatrix} \text{diag}\{\bar{q}_j^{-1}\} & 0_{22 \times 13} \\ 0_{13 \times 22} & \text{diag}\{\bar{u}_i^{-1}\} \end{bmatrix}. \quad (31)$$

This metric measures, in relative and dimensionless form, the weighted Euclidean distance between the actual steady-state augmented vector and its observer-generated estimate, normalized with respect to the weighted distance between the actual operating point and the nominal one. Accordingly, the operating area of the observer is defined as the set of all operating points for which the observer satisfies a prescribed steady-state accuracy requirement, i.e.,

$$\mathbb{H}_\varepsilon(\bar{\mathbf{u}}) = \{\tilde{\mathbf{u}} \in \mathbb{H}(\bar{\mathbf{u}}) : \varepsilon_{ss} \leq \varepsilon_{\max}\}, \quad (32)$$

where $\varepsilon_{\max}$ is a positive parameter selected by the observer designer, that defines the admissible error bound. In the present paper, the observer operating area is chosen to be a 13-dimensional hypercube entirely contained in $\mathcal{H}_\varepsilon(\bar{\mathbf{u}})$. Therefore, every operating point inside this hypercube guarantees that the normalized steady-state estimation error remains below the prescribed threshold, and thus the observer exhibits satisfactory steady-state estimation performance throughout the whole region.

To determine u_i^- and u_i^+ ($i = 1, \dots, 13$), the present paper assumes that the operating area is symmetrical around the nominal value. Let $(u_i)_{\min}$ and $(u_i)_{\max}$ ($i = 1, \dots, 13$) be lower and upper hard bounds that cannot be violated during system operation. Furthermore, consider the affine transformation of the form

$$\tau_i(\lambda) = \begin{cases} \frac{1}{(u_i)_{\max} - (u_i)_{\min}} \lambda - \frac{(u_i)_{\min}}{(u_i)_{\max} - (u_i)_{\min}}, & \text{if } (u_i)_{\max} \neq (u_i)_{\min} \\ 1, & \text{if } (u_i)_{\max} = (u_i)_{\min} \end{cases} \quad (33)$$

and its inverse

$$\tau_i^{-1}(\lambda) = ((u_i)_{\max} - (u_i)_{\min})\lambda + (u_i)_{\min}. \quad (34)$$

The transformation in (33) maps any $u_i \in [(u_i)_{\min}, (u_i)_{\max}]$ to a normalized parameter $\varphi_i = \tau_i(u_i)$ such that $\varphi_i \in [0, 1]$. Similarly, the inverse transformation (34) maps any parameter $\varphi_i \in [0, 1]$ to the original range $[(u_i)_{\min}, (u_i)_{\max}]$.

In what follows, the target operating areas will be determined in the normalized parameter space rather than in the original input space. This is due to the fact that the transformation (33) maps the admissible range of each input variable onto the common interval $[0, 1]$, thereby eliminating the influence of heterogeneous units and unequal numerical ranges. After normalization, all coordinates become dimensionless and directly comparable, while the resulting operating areas retain exact correspondence with the original input space through the inverse transformation (34). Consequently, distances, volumes, and symmetric margins admit a clearer geometric interpretation, and the construction of the operating areas becomes more consistent across all input dimensions. Let $\varphi_i = \tau_i(u_i)$, $\bar{\varphi}_i = \tau_i(\bar{u}_i)$, $\varphi_i^- = \tau_i(u_i^-)$, $\varphi_i^+ = \tau_i(u_i^+)$ ($i = 1, \dots, 13$) and $\bar{\varphi} = [\bar{\varphi}_1 \dots \bar{\varphi}_{13}]^T$. The normalized operating area bounds will be selected to be of the form

$$\varphi_i^-(\bar{\varphi}) = \max\{(1-a)\bar{\varphi}_i, 0\}, \quad (35)$$

$$\varphi_i^+(\bar{\varphi}) = \min\{(1+a)\bar{\varphi}_i, 1\}, \quad (36)$$

where a is an expansion factor under the constraint $a \in [0, 1]$.

Clearly, the exact evaluation of the worst-case steady-state estimation error over the entire operating region leads to a nonconvex global optimization problem in 13 variables. To simplify operating region determination procedure, a randomized approximation is adopted. For a fixed trial value of the relative expansion factor a , independent random relative perturbations of the nominal input are generated in the interval $[1-a, 1+a]$ for each input component and are then saturated to the globally admissible bounds. Each resulting input vector defines a candidate final steady operating point for the nonlinear process. The corresponding normalized steady-state estimation error is evaluated, and the maximum over the random sample is used as an empirical approximation of the worst-case error. The value of a is accepted if no sampled operating point violates the prescribed steady-state accuracy bound. To estimate the maximum value of a for which no sampled operating points violate the prescribed steady-state accuracy bound, an interval division algorithm is employed. Assume that there exist $a_{\min}$ and $a_{\max}$, with $a_{\min} < a_{\max}$ such that when $a = a_{\min}$, no sampled operating point violates the prescribed steady-state accuracy bound (i.e., a is acceptable), whereas when $a = a_{\max}$, there exist operating points violating the prescribed steady-state accuracy bound (i.e., a is not acceptable). Furthermore, define the Boolean function

$$\Phi(a) = \begin{cases} 1, & \text{if } a \text{ is acceptable} \\ 0, & \text{if } a \text{ is not acceptable} \end{cases}. \quad (37)$$

Assuming monotonic behavior, i.e., if a given value of a is acceptable, then every smaller value is also acceptable, the admissible set of values is an interval of the form $[a_{\min}, a^*]$, where a^* is the largest admissible value to be determined. Based on the above definitions, a six-step algorithm (S0 to S5) of the following form may be implemented (Algorithm 2).

Algorithm 2. Interval division algorithm.

S0: Set $a_L = a_{\min}$ and $a_U = a_{\max}$.

S1: If $a_U - a_L \leq \varepsilon$ then go to S5. If $a_U - a_L > \varepsilon$, proceed to S2.

S2: Compute the midpoint $a_M = (a_U + a_L)/2$.

S3: If $\Phi(a_M) = 1$, set $a_L = a_M$, otherwise, set $a_U = a_M$.

S4: Go to S1.

S5: Return the estimate of a^* as $\hat{a}^* = (a_U + a_L)/2$

After determining $\hat{a}^*$, the normalized operating bounds can be determined through (35) and (36) which can be mapped to the original input space through (34). Note that the operating area determination procedure described above can be interpreted as a randomized feasibility test. If $p_i(a)$ denotes the probability under the adopted sampling procedure that a random perturbation of relative size a yields an estimation error above the admissible threshold, then the probability that such a violating point is missed by N independent samples is $(1 - p_i(a))^N$. Hence, if N is selected to be large enough, a^* can be estimated with high confidence.

3.4. Observer-Bank Construction Algorithm

As already mentioned, the above procedure must be repeated for a sufficiently large number of points to ensure that the desired area of inputs is covered by target operating areas that satisfy the dense web principle (see [23]). Let

$$\tilde{\mathbb{H}}_d = \times_{i=1}^{13} [0,1] \quad (38)$$

be the desired area of normalized inputs. For a normalized nominal operating point $\bar{\Phi}^{(\rho)} = [\bar{\varphi}_1^{(\rho)} \ \dots \ \bar{\varphi}_{13}^{(\rho)}]^T$, where $\rho = 1, \dots, \mu$, let $\tilde{\mathbb{H}}_\varepsilon(\bar{\Phi}^{(\rho)}) \subseteq \tilde{\mathbb{H}}_d$ denote the corresponding target operating area and let $\mathfrak{Z}^{(\rho)}$ denote the corresponding observer. The problem considered here is the determination of a finite family of operating points and corresponding observers, i.e., to determine an appropriate finite μ such that

$$\bigcup_{\rho=1}^{\mu} \tilde{\mathbb{H}}_\varepsilon(\bar{\Phi}^{(\rho)}) \subseteq \tilde{\mathbb{H}}_d. \quad (39)$$

Equivalently, if

$$\mathbb{L}_N = \bigcup_{\rho=1}^N \tilde{\mathbb{H}}_\varepsilon(\bar{\Phi}^{(\rho)}) \quad (40)$$

denotes the region covered after the determination of N observers, then the uncovered region is

$$\mathbb{J}_N = \tilde{\mathbb{H}}_d \setminus \mathbb{L}_N \quad (41)$$

and the objective is to construct observers until $\mathbb{J}_N = \emptyset$.

The proposed procedure is adaptive, in the sense that each new operating point is selected from the geometry of the uncovered part of $\tilde{\mathbb{H}}_d$. The first operating point is chosen at the center of the desired input region, namely

$$\bar{\Phi}^{(1)} = \frac{\bar{\Phi}_{\min} + \bar{\Phi}_{\max}}{2}, \quad (42)$$

where $\bar{\Phi}_{\min} = \mathbf{0}_{13 \times 1}$, $\bar{\Phi}_{\max} = \mathbf{1}_{13 \times 1}$ and where $\mathbf{0}_{a \times b}$ and $\mathbf{1}_{a \times b}$ are $a \times b$ constant matrices containing only zeros and ones, respectively.

For this normalized operating point, the nominal steady-state values of the nonlinear model are computed; the corresponding observer is tuned by the algorithm presented previously and the associated target operating area is determined. After N observers have been determined, an exact coverage test is performed with respect to the family of

already computed target operating areas. For each input/coordinate in the 13-dimensional space, define the set of breakpoints

$$\mathbb{G}_i^{(N)} = \{0, 1\} \cup \bigcup_{k=1}^N \left\{ (\varphi_i)_{\min}^{(k)}, (\varphi_i)_{\max}^{(k)} \right\}. \quad (43)$$

By ordering the elements of $\mathbb{G}_i^{(N)}$, the interval $[0, 1]$ is partitioned into elementary one-dimensional intervals. The Cartesian product of these partitions induces a finite decomposition of $\tilde{\mathbb{H}}_d$ into elementary 13-dimensional boxes. Since all boundaries of the already computed target operating areas belong to this partition, each elementary box is either fully covered or fully uncovered. Hence, the uncovered region $\mathbb{J}_N$ can be represented exactly as the union of the uncovered elementary boxes. If $\mathbb{J}_N \neq \emptyset$, a new operating point must be selected inside the uncovered region. To this end, a box-search procedure is employed. Let

$$\mathbb{V} = \times_{i=1}^{13} \left[(b_i)_{\min}, (b_i)_{\max} \right] \subseteq \tilde{\mathbb{H}}_d \quad (44)$$

be a candidate box, i.e., one of the uncovered elementary boxes. Let

$$\text{vol}(\mathbb{V}) = \prod_{i=1}^{13} \left((b_i)_{\max} - (b_i)_{\min} \right) \quad (45)$$

be the 13-dimensional volume of the elementary box $\mathbb{V}$. Let $\mathbb{V}^* = \times_{i=1}^{13} \left[(b_i)_{\min}^*, (b_i)_{\max}^* \right]$ be the box with the maximum volume. The next operating point is then selected as the center of this box, namely

$$\bar{\boldsymbol{\varphi}}^{(N+1)} = \frac{\mathbf{b}_{\min}^* + \mathbf{b}_{\max}^*}{2}, \quad (46)$$

where

$$\mathbf{b}_{\min}^* = \left[(b_1)_{\min}^* \quad \cdots \quad (b_{13})_{\min}^* \right]^T, \quad \mathbf{b}_{\max}^* = \left[(b_1)_{\max}^* \quad \cdots \quad (b_{13})_{\max}^* \right]^T,$$

and the procedure is repeated. Clearly, if more than one uncovered elementary box has the maximum volume, then the center of any one of them may be selected. It is important to mention that since a newly determined target operating area $\tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(\rho)})$ may overlap with, or even fully contain, previously computed target operating areas, the iterative construction of the observer bank may introduce redundant observer–area pairs. To avoid unnecessary observers, an exact elimination step may be performed after each update of the family of target operating areas. More precisely, if for some $k \in \{1, \dots, N\}$ it holds that

$$\tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(k)}) \subseteq \bigcup_{\substack{j=1 \\ j \neq k}}^N \tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(j)}), \quad (47)$$

then $\tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(k)})$ is redundant and may be removed together with the corresponding observer $\mathfrak{S}^{(k)}$. This elimination does not alter the covered region, since equality

$$\bigcup_{j=1}^N \tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(j)}) = \bigcup_{\substack{j=1 \\ j \neq k}}^N \tilde{\mathbb{H}}_\varepsilon(\bar{\boldsymbol{\varphi}}^{(j)}) \quad (48)$$

remains valid after the removal of any redundant target operating area.

Algorithmically, the proposed observer-bank construction can be interpreted as a problem-specific adaptive geometric-covering procedure over the normalized input domain. It follows the general idea of adaptive covering, where the covering elements are generated sequentially and their size is determined from local information (see [51]), but differs from classical geometric set-cover formulations (see [52]), since the candidate target operating areas are not known a priori. Instead, each covering set is produced only after the corresponding nonlinear steady-state computation, local linearization, observer tuning, and target-area validation have been completed. Since the accepted target

operating areas are axis-aligned hyperrectangles, computing the covered region, the uncovered region, and the redundancy condition reduce to union and containment operations for boxes, which are related to Klee-type measure problems (see [53]). The selection of the next nominal operating point as the center of the largest uncovered elementary box introduces a greedy largest gap mechanism, related to largest empty box search problems (see [54]). The complete procedure can be summarized in a seventeen-step procedure (S0 to S16) as follows (Algorithm 3).

Algorithm 3. Observer-bank construction algorithm with elimination of redundant target operating areas.

- S0:** Define $\mathbb{H}_d$. Set $\bar{\varphi}^{(1)} = (\bar{\varphi}_{\min} + \bar{\varphi}_{\max})/2$ and $N = 1$.
- S1:** For the current operating point $\bar{\varphi}^{(N)}$ compute the corresponding nominal steady-state values of the nonlinear model.
- S2:** Construct the local linear approximant and determine the corresponding observer $\mathfrak{Z}^{(N)}$.
- S3:** Determine the target operating area $\tilde{\mathbb{H}}_\varepsilon(\bar{\varphi}^{(N)})$ of $\mathfrak{Z}^{(N)}$.
- S4:** Update the current family of target operating areas by appending $\tilde{\mathbb{H}}_\varepsilon(\bar{\varphi}^{(N)})$.
- For each $k \in \{1, \dots, N\}$, check whether relation (47) holds true. If it does, eliminate
- S5:** $\tilde{\mathbb{H}}_\varepsilon(\bar{\varphi}^{(k)})$ together with the corresponding observer $\mathfrak{Z}^{(k)}$.
- S6:** Repeat S5 until no further target operating areas can be eliminated. Reindex the remaining observer–area pairs and denote their total number again by N .
- S7:** For each coordinate $i \in \{1, \dots, 13\}$, define the set of breakpoints in (43).
Arrange the elements of $\mathbb{G}_i^{(N)}$ in increasing order and form the one-dimensional intervals
- S8:**
$$\mathbb{S}_{i,r_i}^{(N)} = [\gamma_{i,r_i-1}^{(N)}, \gamma_{i,r_i}^{(N)}]; r_i = 1, \dots, v_i, \quad (49)$$
where v_i denotes the number of elementary one-dimensional intervals induced along the i -th coordinate.
- S9:** Construct the elementary 13-dimensional boxes $\mathbb{E}_{r_1, \dots, r_{13}}^{(N)} = \times_{i=1}^{13} \mathbb{S}_{i,r_i}^{(N)}$.
Determine the set of uncovered elementary boxes
- S10:**
$$\mathbb{P}^{(N)} = \left\{ (r_1, \dots, r_{13}) : \mathbb{E}_{r_1, \dots, r_{13}}^{(N)} \cap \bigcup_{j=1}^N \tilde{\mathbb{H}}_\varepsilon(\bar{\varphi}^{(j)}) = \emptyset \right\}. \quad (50)$$
- S11:** If $\mathbb{P}^{(N)} = \emptyset$, go to S16.
- S12:** For each $(r_1, \dots, r_{13}) \in \mathbb{P}^{(N)}$, compute $\text{vol}(\mathbb{E}_{r_1, \dots, r_{13}}^{(N)})$ through (45).
Select an uncovered elementary box of maximum volume, i.e., choose
- S13:**
$$(r_1^*, \dots, r_{13}^*) = \arg \max_{(r_1, \dots, r_{13}) \in \mathbb{P}^{(N)}} \text{vol}(\mathbb{E}_{r_1, \dots, r_{13}}^{(N)}). \quad (51)$$
- Set the next operating point equal to
- S14:**
$$\bar{\varphi}^{(N+1)} = \left[\left(\gamma_{i,r_1^*-1}^{(N)} + \gamma_{i,r_1^*}^{(N)} \right) / 2 \quad \dots \quad \left(\gamma_{i,r_{13}^*-1}^{(N)} + \gamma_{i,r_{13}^*}^{(N)} \right) / 2 \right]^T.$$
- S15:** Set $N = N + 1$ and go to S1.
- S16:** Terminate.
-

The above procedure constructs a finite observer bank whose target operating areas cover the desired area of inputs, while the elimination step removes redundant observer–area pairs without affecting the covered region.

3.5. Switching Between Observers

As already mentioned, the observer matrices depend upon the operating point of the system. Consequently, for the finite family of nominal operating points determined in the previous subsection, a corresponding bank of observers $\mathfrak{Z}^{(1)}, \mathfrak{Z}^{(2)}, \dots, \mathfrak{Z}^{(\mu)}$ is obtained, where observer $\mathfrak{Z}^{(i)}$ is associated with nominal operating point $\bar{\mathbf{q}}^{(i)}$ and target operating area $\mathbb{H}_\varepsilon(\bar{\mathbf{u}}^{(i)})$ ($i = 1, \dots, \mu$). For proper operation of the soft sensor, an appropriate switching mechanism is required, so that the observer used for estimation remains compatible with the current operating condition of the nonlinear process. The switching policy adopted here is based on the overlap of target operating areas and on the convergence of the measurable outputs to a steady-state value corresponding to inputs belonging in the overlapping parts of the target operating areas. Let $\mathfrak{Z}^{(i)}$ be the currently active observer and $\mathfrak{Z}^{(j)}$ be a candidate observer. If

$$\mathbb{H}_\varepsilon(\bar{\mathbf{u}}^{(i)}) \cap \mathbb{H}_\varepsilon(\bar{\mathbf{u}}^{(j)}) \neq \emptyset, \quad (52)$$

then both observers are valid in the overlap region, and therefore a transition from $\mathfrak{Z}^{(i)}$ to $\mathfrak{Z}^{(j)}$ is admissible whenever the process approaches this region. Define the overlap region

$$\mathbb{H}_\varepsilon^{(i,j)} = \mathbb{H}_\varepsilon(\bar{\mathbf{u}}^{(i)}) \cap \mathbb{H}_\varepsilon(\bar{\mathbf{u}}^{(j)}). \quad (53)$$

By construction, every operating point in $\mathbb{H}_\varepsilon^{(i,j)}$ is admissible for both observers. Therefore, switching should take place when the process approaches $\mathbb{H}_\varepsilon^{(i,j)}$. Since the measurable output vector is available in real time, the switching decision is based on the measurable output proximity to the overlap region. Let

$$\mathbf{y}(k) = \mathbf{C}_d \mathbf{q}(k) \quad (54)$$

denote the measurable output vector at sampling instant k . Furthermore, let $\bar{\mathbf{y}}^{(i)} = \mathbf{C}_d \bar{\mathbf{q}}^{(i)}$ be the nominal measurable output vector associated with the current observer $\mathfrak{Z}^{(i)}$ and $\tilde{\mathbf{y}} = \mathbf{C}_d \tilde{\mathbf{q}}$ be the steady-state value of the measurable output vector corresponding to the steady-state value of the inputs belonging to the overlap region. The following time-dependent convergence metric is then introduced:

$$\varepsilon_c(k) = \frac{\|\mathbf{y}(k) - \tilde{\mathbf{y}}\|_2}{\|\bar{\mathbf{y}}^{(i)} - \tilde{\mathbf{y}}\|_2} \times 100\%. \quad (55)$$

Switching between observers will take place whenever the convergence metric reaches a threshold $\varepsilon_s \in \mathbb{R}^+$, i.e., when $\varepsilon_c(k) \leq \varepsilon_s$.

4. Performance of the Soft Sensor Scheme

To demonstrate the performance of the proposed soft sensor scheme, a two-stage approach is adopted. First, the system performance is evaluated through a series of computational experiments. In this phase, for a prescribed input area of the water distribution network, the observer-bank construction algorithm presented in Section 3.4 is implemented, and its performance is assessed for a representative transition between an initial and a final point within the specified input area.

In the second stage, using the observers and target operating areas identified during the simulation phase, a simulation-supported edge runtime validation approach is employed to evaluate the operation of the proposed scheme under realistic industrial communication network conditions. In this case, measurement data are generated by a

computer-based simulation of the network operation. The architecture is based on a Node-RED edge runtime, which acquires the generated “measurements”, executes a bank of discrete-time observers, and forwards to the cloud both the state-variable estimates produced by the observer bank and the system output “measurements” used to trigger a supervisory switching signal.

4.1. Model Parameters and Input Area Selection

Let $L_j = 300[\text{m}]$, $D_j = 0.1[\text{m}]$, $\varepsilon_j = 0.35[\text{mm}]$ ($j = 1, \dots, 11$), $g = 9.81[\text{m/s}^2]$, $\rho = 999.1[\text{Kg/m}^3]$, $\mu = 0.0011[\text{Pa s}]$, $k_b = 0.1[\text{h}^{-1}]$, $b = 1200[\text{m/s}]$, $\lambda_1 = 0.11[-]$, $\lambda_2 = 68[-]$ and $\varepsilon_{ss, \max} = 5[\%]$. Regarding the sampling period, it is assumed to be $T_s = 15[\text{s}]$ which is a valid assumption for typical primary water distribution networks. Considering the constant flow direction constraint imposed in Section 2.2 and to identify admissible nominal operating conditions, a numerical testing procedure will be implemented to explore the space of nominal input values. Clearly, chlorine concentration dynamics do not affect the water flow conditions. Consequently, they can be omitted from flow reversal analysis. Let $\bar{u}_1 \in [28, 32][\text{m}]$, $\bar{u}_j \in [4.5, 5.5][\text{m}]$ ($j = 2, \dots, 7$), $\bar{u}_l \in [0, 0.0003][\text{m}^3/\text{s}]$ ($l = 9, 10, 13$) and $\bar{u}_m \in [0, 0.003][\text{m}^3/\text{s}]$ ($m = 11, 12$). The objective of the test was to determine combinations of nominal inputs that produce admissible / positive nominal flow and nodal pressure values. More specifically, for each sampled input vector, the corresponding nominal solution, let $\bar{x}_r = [\bar{x}_1 \ \dots \ \bar{x}_{11}]^T$, was computed and assessed with respect to the positivity condition $\bar{x}_j \geq 0$ ($j = 1, \dots, 11$). Input vectors satisfying this condition were classified as non-flow-reversal operating points, whereas samples producing at least one negative nominal flow value were classified as flow-reversal operating points. In the present approach, the input space is examined using a sampling-based search strategy. A large set of input combinations is generated within the prescribed hyper-rectangular domain, and additional deterministic samples corresponding to the vertices of the domain and its midpoint are also included. The search set combined every vertex of the 12-dimensional hyperrectangle, a 50,000-point Latin-hypercube design in its interior as well as 300 randomized points on each lower and upper boundary face. The numerical search was performed over a total of 61,298 candidate nominal input vectors. All evaluated samples resulted in solver-valid nominal solutions, indicating that the reduced scalar evaluation procedure was numerically robust over the considered input domain. No sample was rejected due to solver failure or excessive residual error. Among the combinations, 6044 preserved all reference flow directions and 55,254 contained at least one reversed flow. The corresponding test-set frequencies were 9.860% and 90.140%, respectively, while no sampled combination belonged to the near-zero class. The reversals were confined to the downstream part of the network. The nominal flows of x_1 to x_5 (see Figure 2) remained positive for every evaluated combination. The reversal rates of x_6 and x_7 were 19.084% and 19.159%, respectively, and the rates of x_8 and x_9 were 12.418% and 15.337%. The most sensitive branches were x_{10} and x_{11} , which reversed in 40.218% and 40.279% of the valid combinations. The number of simultaneously reversed flows remained limited (see Figure 3). A single reversed branch occurred in 27,491 combinations, corresponding to 44.848% of all valid samples, while two reversed branches occurred in 21,666 combinations, corresponding to 35.345%. Three and four simultaneous reversals occurred in 8.831% and 1.116% of the valid combinations, respectively. Thus, 88.966% of all reversal cases involved only one or two branches, and no sampled combination contained more than four reversed flows.

From the data analysis it can be verified that the identified non-flow-reversal samples provide a feasible set of candidate operating conditions for the design and validation of the proposed observer-based methodology. Furthermore, the numerical point cloud indicates that the admissible point set is irregular, with admissible samples appearing only in restricted coupled regions of the input space. Therefore, the admissible domain cannot be represented by simple independent bounds on the nominal values of the inputs. For this reason and to demonstrate the soft sensor design procedure and performance, a conservative hyper-rectangular subarea will be selected. Without loss of generality, it will be assumed that $\bar{u}_1 = 30[\text{m}]$, $\bar{u}_j = 5[\text{m}]$ ($j = 2, \dots, 7$), $\bar{u}_9 = 0.0002[\text{m}^3/\text{s}]$, $\bar{u}_{10} = 0.0001[\text{m}^3/\text{s}]$, $\bar{u}_{13} = 0.0001[\text{m}^3/\text{s}]$ and $\bar{u}_j \in [0, 0.003][\text{m}^3/\text{s}]$ ($j = 11, 12$). It can be readily verified that in the above regions, no flow reversal occurs. Regarding the range of nominal values of the influent chlorine concentration, it will be assumed that $\bar{u}_8 \in [0, 3][\text{mg/L}]$.

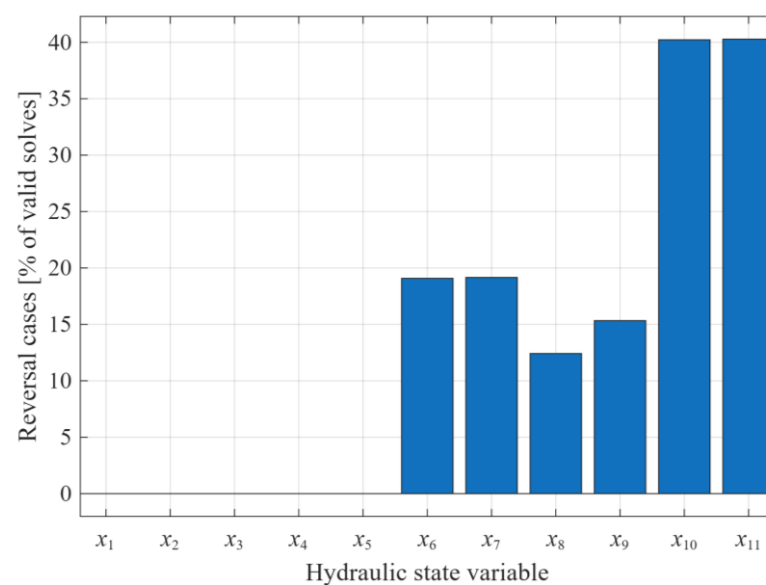


Figure 2. Percentage of combinations producing a reversal in each nominal pipe flow.

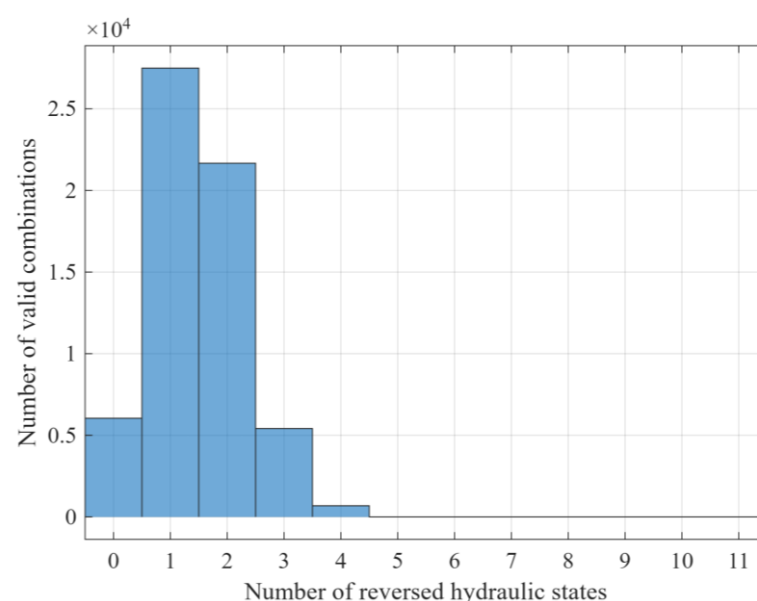


Figure 3. Distribution of the number of reversed nominal pipe flows per input combination.

4.2. Target Operating Areas

In the present subsection, following the results of Section 3, a soft sensor will be designed, toward estimation of the nonmeasurable chlorine concentration variables of the benchmark water distribution network presented in Section 2. Toward implementation of the affine transformations (33) and (34), it clearly holds that $(u_8)_{\min} = 0[\text{mg/L}]$, $(u_{11})_{\min} = 0[\text{m}^3/\text{s}]$, $(u_{12})_{\min} = 0[\text{m}^3/\text{s}]$, $(u_8)_{\max} = 3[\text{mg/L}]$, $(u_{11})_{\max} = 0.003[\text{m}^3/\text{s}]$ and $(u_{12})_{\max} = 0.003[\text{m}^3/\text{s}]$. Regarding implementation of the algorithm, the sampling period is set to $T_s = 15[\text{s}]$; the admissible normalized steady-state estimation error was set to $\varepsilon_{\max} = 5[\%]$ and each candidate target operating area is validated using 5000 sampled operating points generated by Latin-hypercube sampling. The expansion factor of each target operating area was determined by bisection with tolerance of 10^{-6} . The observer-pole search was constrained to real nonnegative discrete-time poles in the interval $[0, 1)$ with a minimum pole spacing of 0.005 and a minimum ordered distance of 0.01 from the plant poles. The free observer-gain parameters were searched in the interval $[-100, 100]$.

Regarding selection of the number of sampled operating points, let p_v denote the probability, under the adopted sampling distribution, that an operating point within a candidate target area violates the prescribed steady-state accuracy threshold. If no violation is observed among N validation samples, the probability of missing a violating point is $(1 - p_v)^N$. Consequently, with confidence $1 - \beta$, the violation probability satisfies $p_v \leq 1 - \beta^{-N}$. For $N = 5000$, zero observed violations correspond to upper bounds of approximately 0.0599% and 0.0921% at confidence levels of 95% and 99%, respectively. Equivalently, a violating subset with probability of at least 0.1% would be detected with approximately 99.33% probability.

Application of the observer-bank construction algorithm presented in Section 3.4 results in a three-dimensional volume coverage of 88.82% after producing a bank of 242 active observers and the corresponding operating areas. From the simulation results it can be verified that the number of active observers may change during algorithm implementation due to elimination of previously evaluated redundant observers and corresponding operating areas. In Figures 4–9 graphical presentations of the coverage is represented for indicative algorithm loops, where it can be verified that the three-dimensional input space (since ten of the inputs are constant) is gradually covered by the new observers and their corresponding operating areas. The evolution of coverage clearly shows the effect of the redundancy-elimination mechanism. In the early iterations, large target operating areas remove previously selected smaller regions.

The evolution of the covered volume shows that the proposed procedure initially captures the broader admissible regions and subsequently resolves the remaining localized gaps. The first observer covers approximately 10.82% of the normalized three-dimensional scheduling domain, whereas the largest individual retained target area, generated at construction Step 8, occupies approximately 30.15% of this domain. The total covered volume reaches approximately 43.98% after eight construction steps, 64.37% after 22 steps, 71.32% after 39 steps, and 77.91% after 88 steps. At Step 274, 242 observers remain active following the elimination of 32 redundant pairs, and their union covers approximately 88.82% of the prescribed domain, leaving 11.18% uncovered. Hence, the increase in coverage from approximately 77.91% to 88.82% requires 186 additional construction steps, indicating that the remaining uncovered set is fragmented into small regions that require highly localized observers. The distribution of the nominal operating points is also informative. The normalized u_8 and u_{11} are broadly centered within their prescribed

ranges, with mean values of approximately 0.548 and 0.527 and median values of approximately 0.547 and 0.541, respectively. By contrast, the normalized u_{12} coordinate has a mean value of approximately 0.746 and a median value of approximately 0.792, while 162 of the 242 retained operating points, corresponding to approximately 66.94%, are located in its upper quarter. These results demonstrate an asymmetric distribution of the operating points and a substantially finer partitioning requirement near the upper part of the selected u_{12} range.

It is important to mention that the evaluation of the target operating areas, including nonlinear steady-state calculation, local linearization, observer-gain tuning, target-operating-area validation, coverage evaluation, and redundancy elimination, is a very computationally demanding stage. Nevertheless, as it is a procedure performed offline and does not enter the real-time execution cycle of the soft sensor, it does not constitute a critical computational limitation for the implementability of the soft sensor.

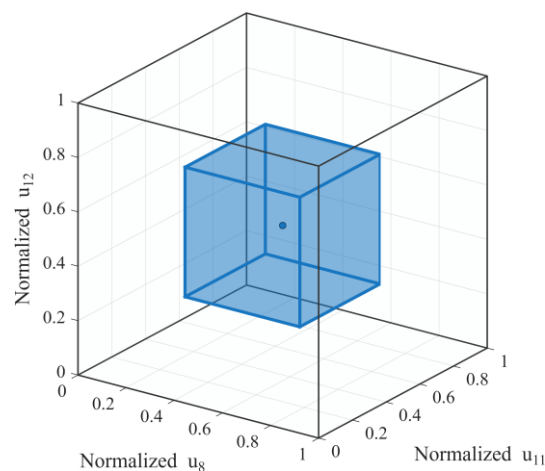


Figure 4. Three-dimensional volume coverage after Step 1.

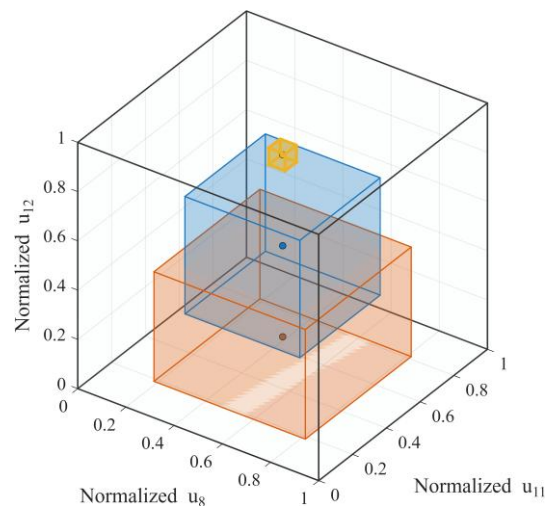


Figure 5. Three-dimensional volume coverage after Step 3.

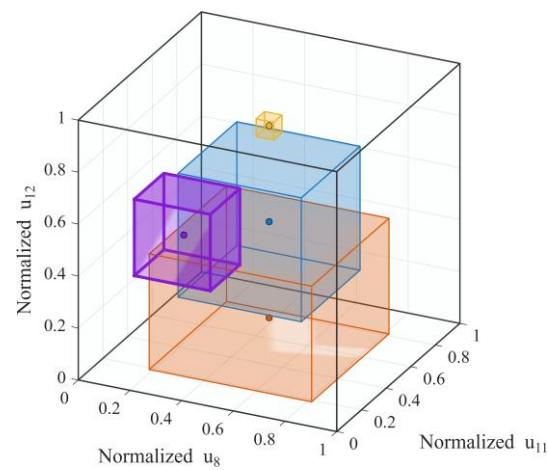


Figure 6. Three-dimensional volume coverage after Step 4.

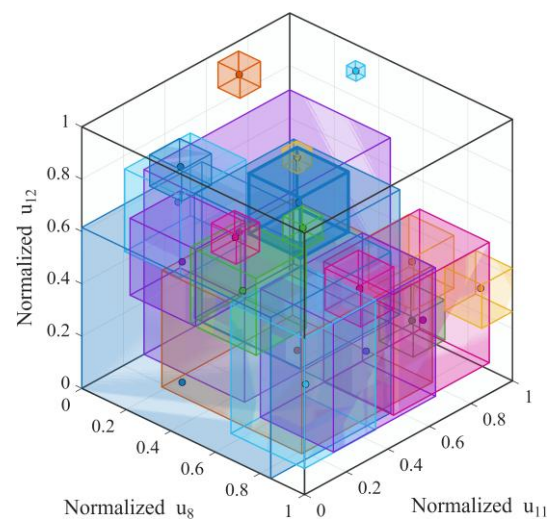


Figure 7. Three-dimensional volume coverage after Step 22.

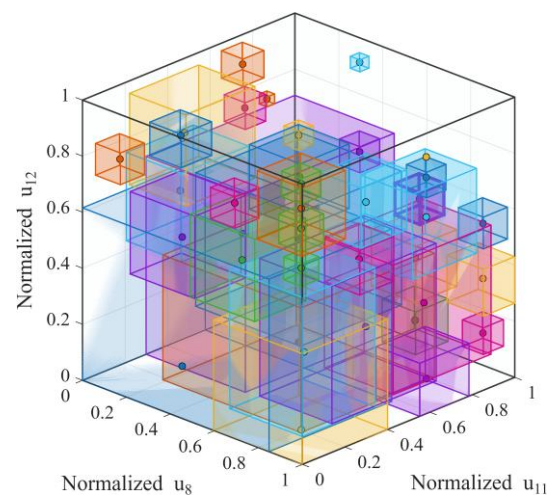


Figure 8. Three-dimensional volume coverage after Step 39.

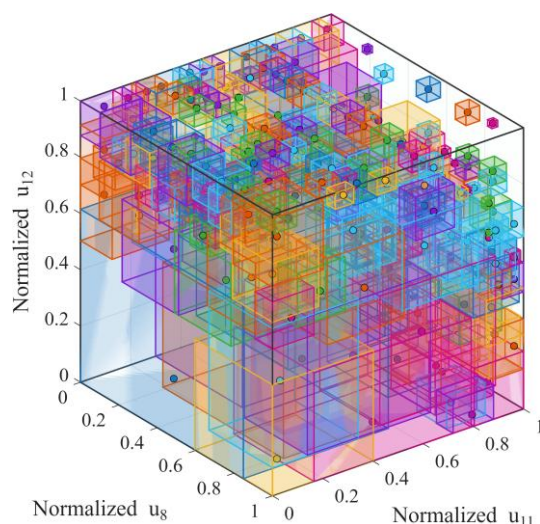


Figure 9. Three-dimensional volume coverage after Step 274.

4.3. Simulation Results

The performance of the switched soft sensor scheme will be evaluated through a hybrid continuous/discrete simulation. The nonlinear hydraulic and chlorine concentration model will be used as the reference process, while the soft sensor will be implemented as a bank of discrete-time linear observers associated with precomputed operating areas. The simulation was initialized at the steady state corresponding to the first selected input point. Therefore, the initial estimation error is set to zero and the subsequent deviations were caused only by the imposed input transitions and by the switching between local observer models. The input sequence started from a point covered only by Observer 8, will pass through overlap regions of consecutive observer operating areas, and will end at a point covered only by Observer 1. Considering the observer tuning results presented in the previous subsection, the selected observer path will be $8 \rightarrow 2 \rightarrow 13 \rightarrow 1$. Each intermediate input point will be selected so that it belongs to the overlap of two consecutive operating areas. That way the currently active observer and the next observer are both valid at the switching point. The physical input values used in the simulation are summarized in Table 1.

Table 1. Input sequence used in the hybrid nonlinear simulation.

Step	Region Type	Observer Membership	u_8 [mg/l]	u_{11} [m ³ /s]	u_{12} [m ³ /s]
1	Unique start	8	2.494287	0.001421	0.001809
2	Overlap	8, 2	0.595551	0.000581	0.000027
3	Overlap	2, 13	2.434529	0.002419	0.000687
4	Overlap	13, 1	2.064902	0.001898	0.001715
5	Unique end	1	0.813750	0.000814	0.002186

The nonlinear model was integrated with a maximum integration step of 1 s, whereas the observer was updated every 15 s. Thus, the observer operated at the sampling period used for the discrete-time soft sensor design, while the continuous-time plant dynamics were resolved on a finer numerical time scale. Switching was enabled using relation (55) setting $\varepsilon_s = 1\%$. After each switch, the absolute estimated chlorine vector was preserved and only its deviation-coordinate representation was changed according to the nominal point of the newly activated observer. Consequently, the observer switching operation did not introduce artificial jumps in the estimated absolute chlorine concentrations.

From Table 1 as well as the simulation results (see Figures 10–31), it can be verified that the simulation involved four input transitions and three observer switches (see Table 2). All four transitions satisfied the prescribed convergence requirement before the maximum segment time. The switches from Observer 8 to Observer 2, from Observer 2 to Observer 13, and from Observer 13 to Observer 1 occurred at 223.75, 473.50, and 759.25 min, respectively. Observer 1 then remained active while the process approached the final point, which belongs exclusively to its target operating area. The complete experiment lasted 1174.75 min, corresponding to approximately 19.58 h. The shortest transition required 223.75 min; the longest required 415.50 min, and the mean transition duration was 293.69 min. Table 2 reports the final measurable-output convergence metric and the normalized state-estimation error at the end of each segment. The final convergence metric remained below 1% in every case, whereas the final normalized estimation error did not exceed 0.156547%.

Table 2. Transition times and switching accuracy, and final estimation accuracy.

Transition	Observer Switch	End Time [min]	Duration [min]	ϵ_c [%]	ϵ_{ss} [%]
1 $\rightarrow$ 2	8 $\rightarrow$ 2	223.75	223.75	0.992441	0.058558
2 $\rightarrow$ 3	2 $\rightarrow$ 13	473.50	249.75	0.999383	0.044245
3 $\rightarrow$ 4	13 $\rightarrow$ 1	759.25	285.75	0.998654	0.035599
4 $\rightarrow$ 5	No change	1174.75	415.50	0.995911	0.156547

The hydraulic trajectories remained compatible with the prescribed flow orientation throughout the complete experiment. The chlorine concentration estimates closely followed the nonlinear model trajectories over the complete sequence. The mean of the 22 normalized error values was 0.71%, and 16 of the 22 chlorine states exhibited an error below 1%. The largest value was 2.27% for state x_{35} . Larger errors were concentrated in states x_{34} , x_{35} and x_{37} , whose trajectories display a small transient lag during the final and longest transition. For the six measured chlorine states x_{20} , x_{28} , x_{30} , x_{32} , x_{36} , and x_{38} , the normalized root-mean-square error (RMSE) remained below 0.3%. When all 22 states and all observer sampling instants were considered jointly, the aggregate RMSE and mean absolute error were 0.01177[mg/L] and 0.0042[mg/L], respectively. The largest absolute estimation error observed during the experiment was 0.128116[mg/L] for state variable x_{17} during the initial transition. Among the measured output states, the maximum absolute error was 0.029208[mg/L] and occurred for x_{20} . The instantaneous normalized error of the complete 22-state estimate reached a maximum of 2.896651% at 812.75 min during the final transition, and subsequently decreased as the process approached the final operating point. The concentration responses therefore show that the switching mechanism selected the prescribed observer sequence without estimate jumps, preserved the assumed flow directions, and maintained accurate reconstruction of both measured and nonmeasurable chlorine concentrations over widely separated operating conditions.

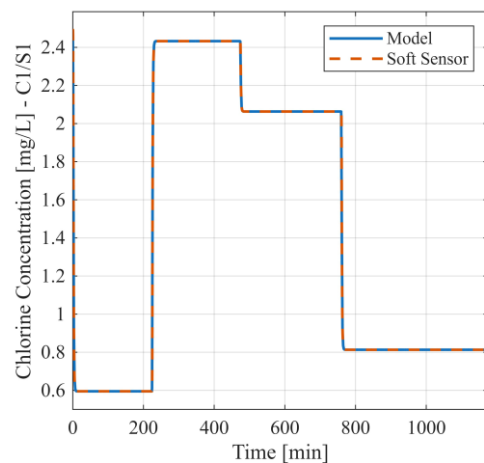


Figure 10. Chlorine concentration estimation in Section 1 of Conduit 1.

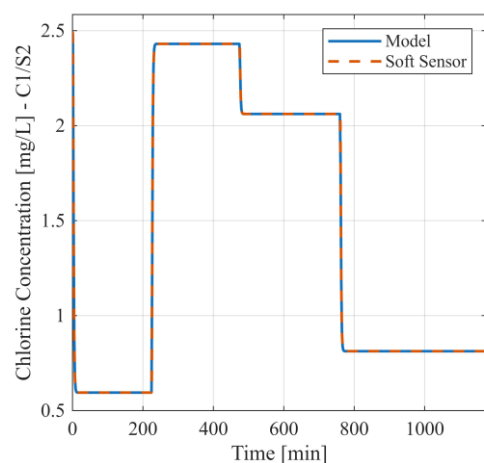


Figure 11. Chlorine concentration estimation in Section 2 of Conduit 1.

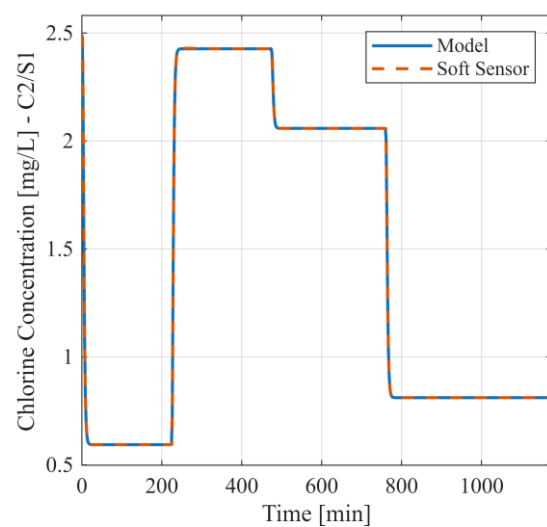


Figure 12. Chlorine concentration estimation in Section 1 of Conduit 2.

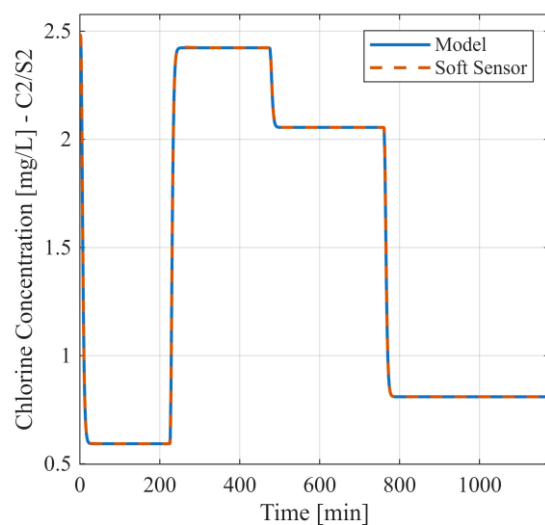


Figure 13. Chlorine concentration estimation in Section 2 of Conduit 2.

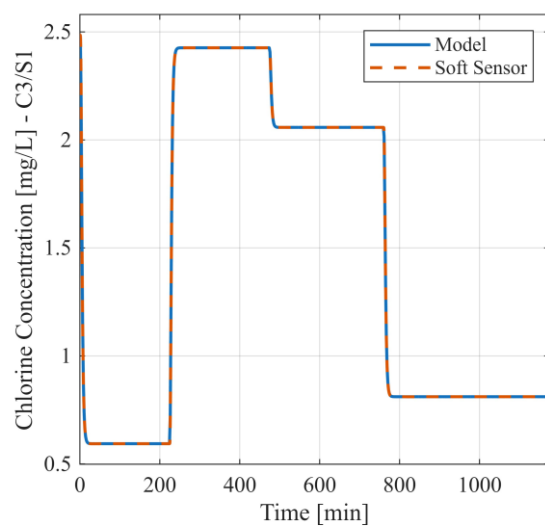


Figure 14. Chlorine concentration estimation in Section 1 of Conduit 3.

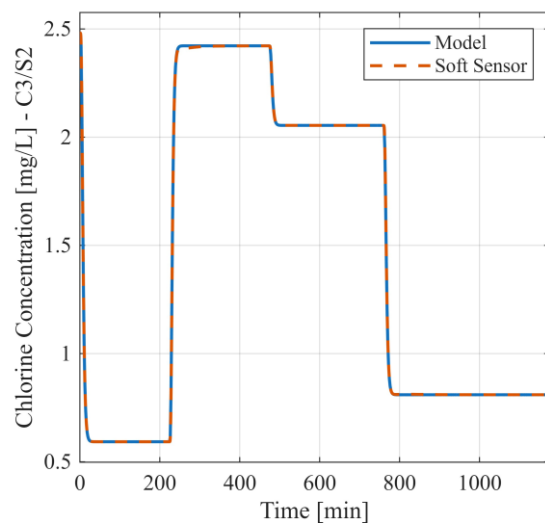


Figure 15. Chlorine concentration estimation in Section 2 of Conduit 3.

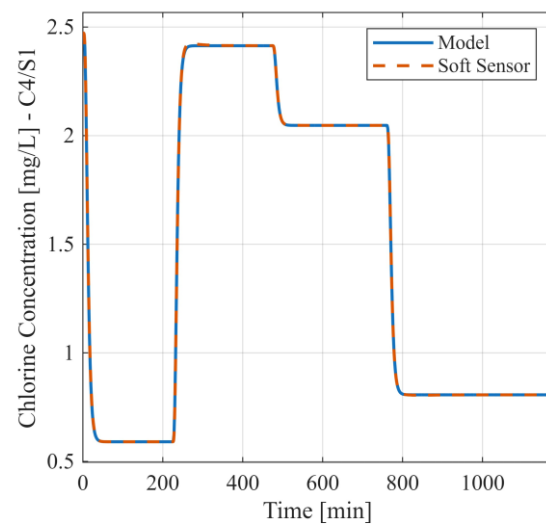


Figure 16. Chlorine concentration estimation in Section 1 of Conduit 4.

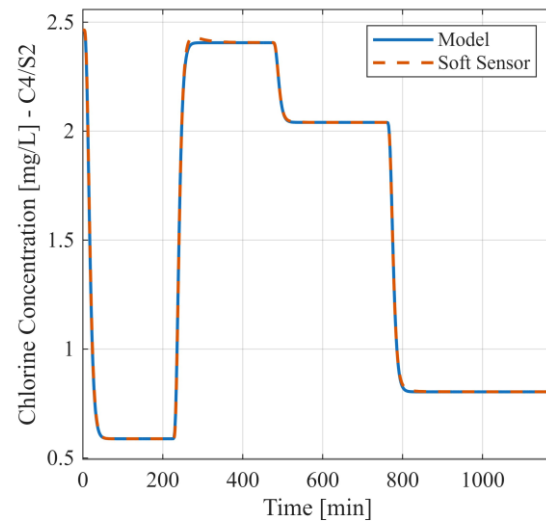


Figure 17. Chlorine concentration estimation in Section 2 of Conduit 4.

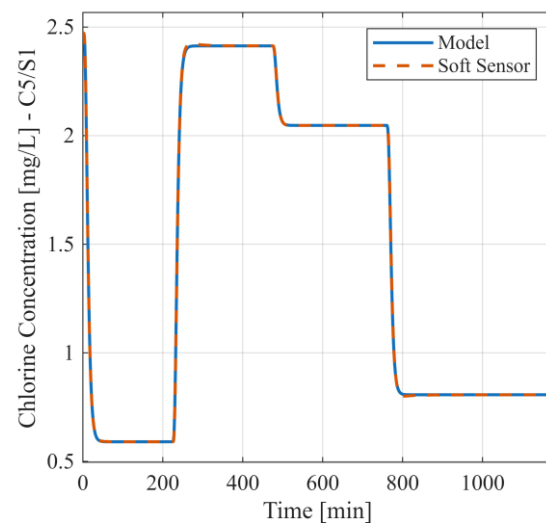


Figure 18. Chlorine concentration estimation in Section 1 of Conduit 5.

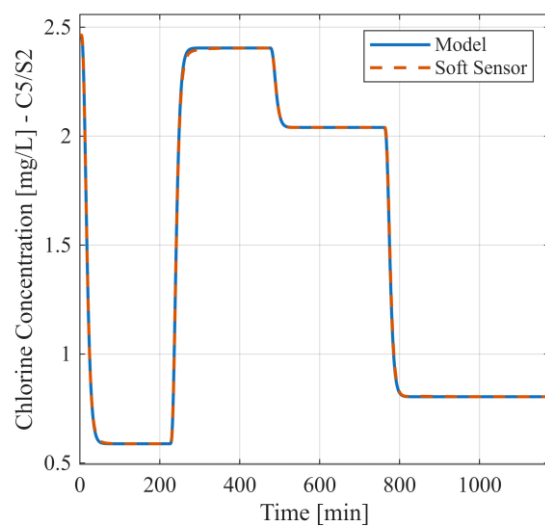


Figure 19. Chlorine concentration estimation in Section 2 of Conduit 5.

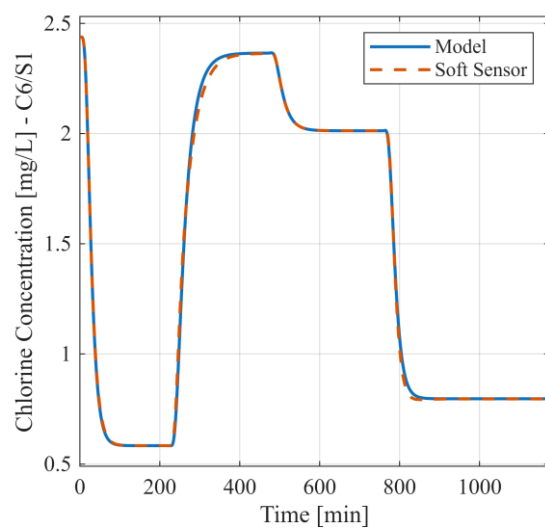


Figure 20. Chlorine concentration estimation in Section 1 of Conduit 6.

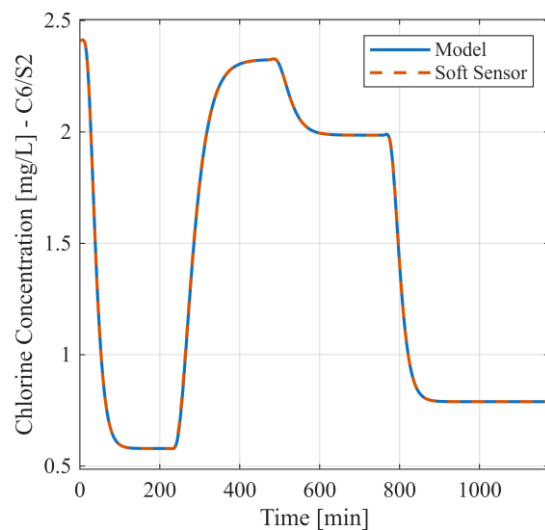


Figure 21. Chlorine concentration estimation in Section 2 of Conduit 6.

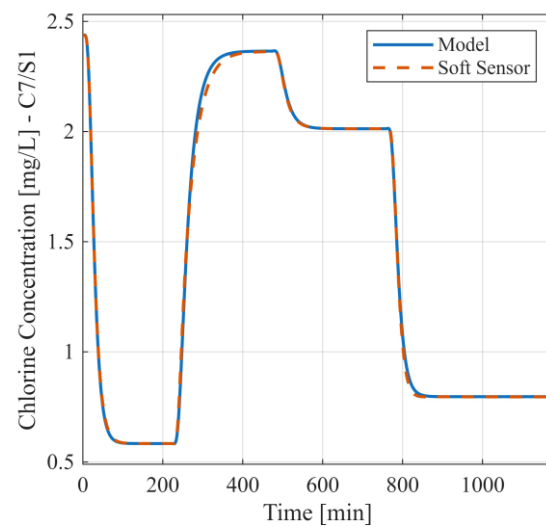


Figure 22. Chlorine concentration estimation in Section 1 of Conduit 7.

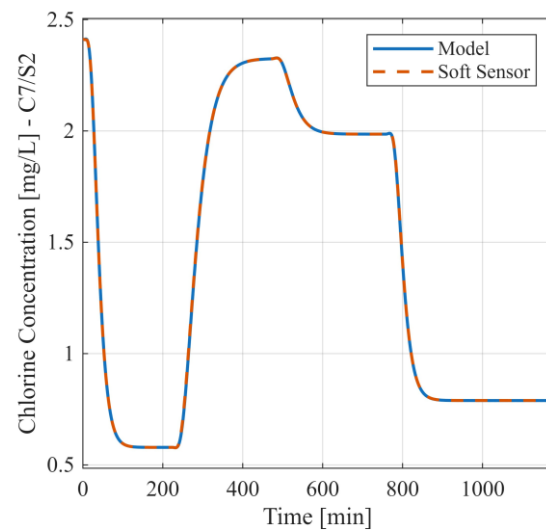


Figure 23. Chlorine concentration estimation in Section 2 of Conduit 7.

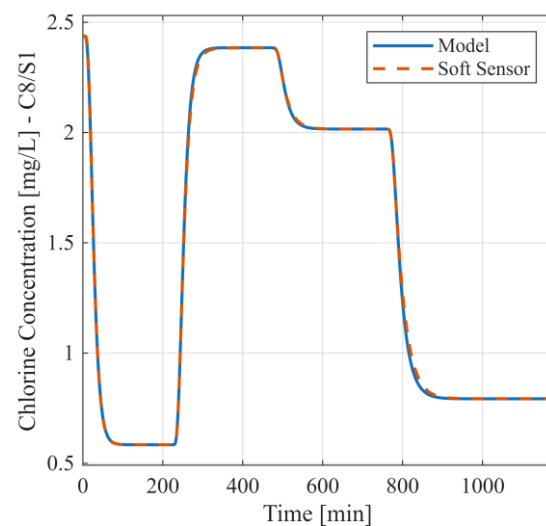


Figure 24. Chlorine concentration estimation in Section 1 of Conduit 8.

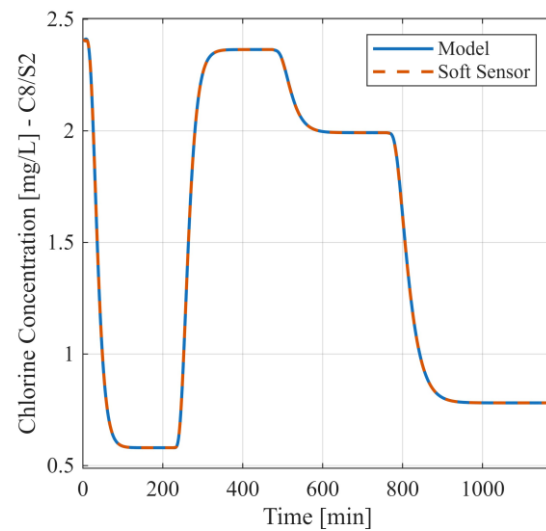


Figure 25. Chlorine concentration estimation in Section 2 of Conduit 8.

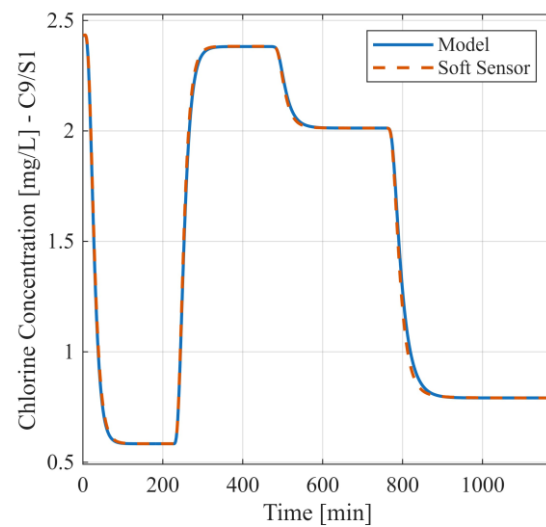


Figure 26. Chlorine concentration estimation in Section 1 of Conduit 9.

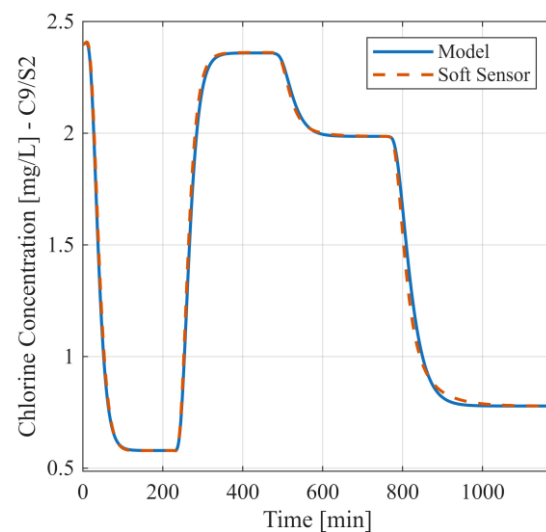


Figure 27. Chlorine concentration estimation in Section 2 of Conduit 9.

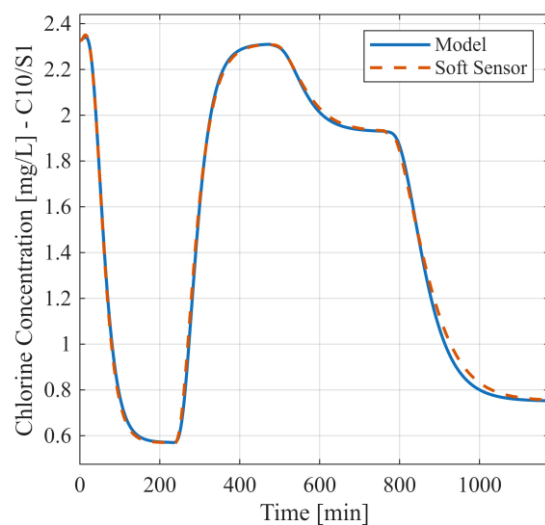


Figure 28. Chlorine concentration estimation in Section 1 of Conduit 10.

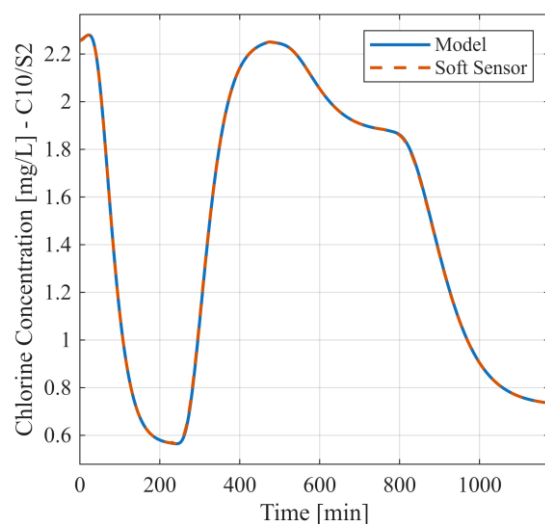


Figure 29. Chlorine concentration estimation in Section 2 of Conduit 10.

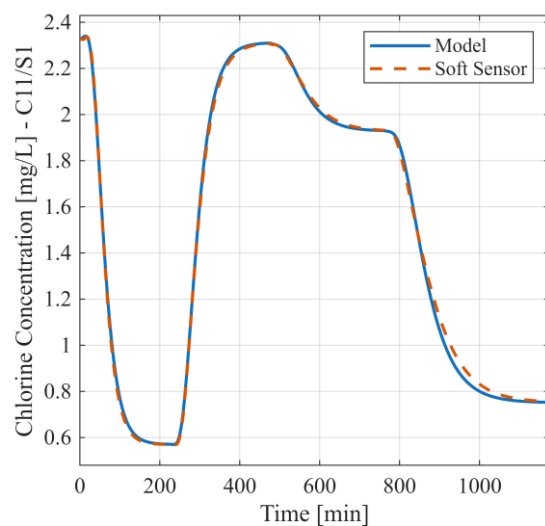


Figure 30. Chlorine concentration estimation in Section 1 of Conduit 11.

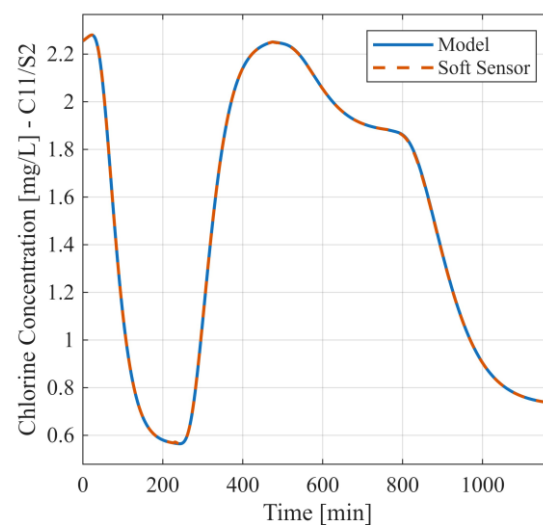


Figure 31. Chlorine concentration estimation in Section 2 of Conduit 11.

Note that extending the simulated trajectory to include a larger part of the observer bank is straightforward, since additional transitions can be introduced by concatenating pairwise-overlapping operating areas and applying the same convergence rules. Because the observer matrices and their target operating areas are computed and validated offline, increasing the path length primarily increases the total simulation duration and number of switching events, without materially affecting the per-sample computational burden. The estimation performance is governed by the locally validated observers and overlap conditions rather than by the total number of observers traversed.

It is important to mention that a comparison with a direct single-step transition from the initial point directly to the final, using the fixed Luenberger observer of Area 8 confirmed the improved transient robustness of the proposed multistep strategy. The multistep procedure reduced the peak normalized error by 16.95% and the maximum chlorine-estimation error by 48.35%, while providing lower maximum errors for 15 of the 22 concentration states.

Regarding the performance of the soft sensor scheme in the presence of uncertainties to model parameters, a sensitivity analysis was performed by perturbing the pipe roughness ε_i ($i = 1, \dots, 11$), the friction parameters λ_1 and λ_2 , and bulk chlorine-decay coefficient k_b by $\pm 2.5\%$ and $\pm 5\%$, while keeping the soft sensor and switching logic unchanged. All perturbed cases completed the prescribed observer sequence with three switches, and every transition satisfied the convergence criterion. The estimation accuracy was most sensitive to λ_1 . A $\pm 5\%$ variation increased the maximum normalized RMSE, peak normalized error, and maximum chlorine-error norm by 21.60%, 21.68%, and 23.51%, respectively. Variations in roughness and λ_2 had considerably smaller effects, whereas k_b mainly influenced the terminal accuracy, with a maximum increase of 17.49% in the final normalized error. Pipe roughness had a moderate but systematic influence on the transient estimation accuracy. A 5% roughness variation increased the maximum normalized RMSE, peak normalized error, and maximum chlorine-error norm by 3.24%, 4.02%, and 4.92%, respectively. Conversely, a -5% variation reduced the peak error but increased the final normalized error and the time spent above the prescribed error threshold.

5. Implementation

The proposed switching-observer soft sensor was implemented in a Node-RED runtime to verify that the observer bank and the switching supervisor can be executed in

a deployable edge-oriented environment. The implementation follows a distributed edge-supervisory architecture. The local layer (see Figure 32) executes the discrete-time observers, whereas the supervisory layer (see Figure 33) evaluates the operating-area information and sends the active observer identifier back to the local layer. In the validation setup, deterministic telemetry generated by the MATLAB benchmark simulation was relayed through MQTT so that the Node-RED implementation received the same input sequence as that used in the computational study of Section 4. This made the implementation test reproducible and enabled direct comparison between the Node-RED estimates and the MATLAB reference trajectories.

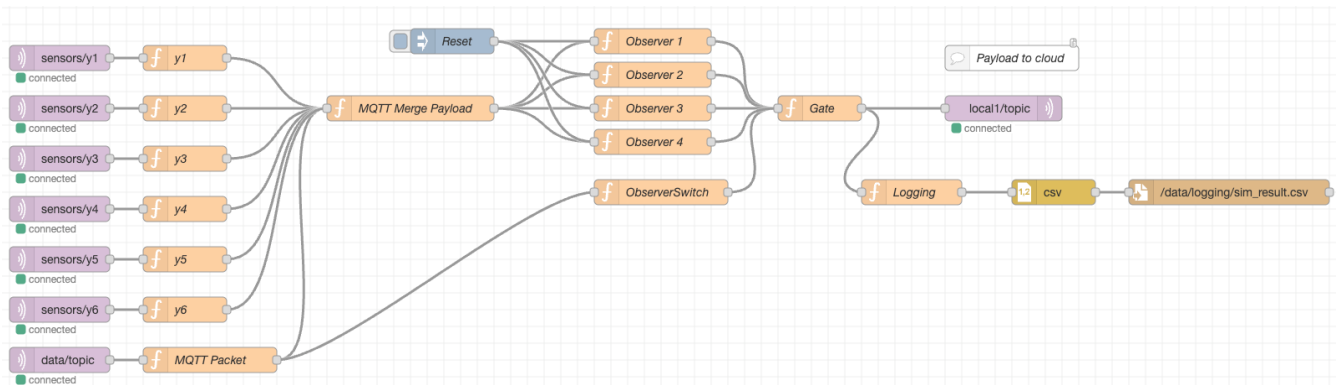


Figure 32. Local Node-RED observer-bank execution flow.

At each sampling instant, the local flow receives the process input vector and the measured output vector. The six measured outputs correspond to the chlorine concentrations measured at the outlet sections of Conduits 2, 6, 7, 8, 10 and 11. The MQTT input nodes subscribe to the topics sensors/y1, sensors/y2, sensors/y3, sensors/y4, sensors/y5 and sensors/y6 for the measured outputs and to data/topic for the supervisory information returned by the supervisory layer. A merge function assembles the latest values into a single payload containing the input vector, the measured output vector, the active observer identifier, and a timestamp. This payload is forwarded to the observer bank.

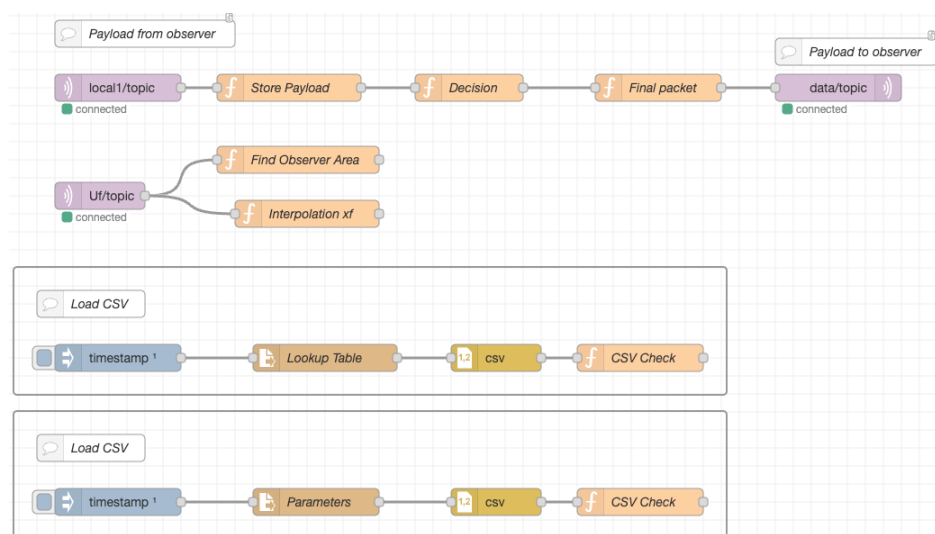


Figure 33. Node-RED edge-cloud implementation architecture.

Each observer is implemented as a JavaScript function node that executes the discrete-time Luenberger-type observer equation derived in Section 3.1. The observer functions run in parallel and produce candidate estimates of the chlorine-state vector. The gate

function (see Figure 34) performs telemetry arbitration. It compares the identifier of each candidate observer with the active observer identifier received from the supervisory layer, rejects non-active estimates, and publishes only the selected estimate. The outgoing packet is transmitted through the topic local1/topic and contains the timestamp, active observer identifier, selected estimate, and the corresponding measured-output data. This structure preserves parallel observer execution while enforcing a single output stream.

```

1  let active = flow.get("activeObserver") || 8;
2
3  if (msg.observer !== active) {
4    return null;
5  }
6
7  const xestArr = Array.from(
8    { length: 22 },
9    (_, i) => msg.payload.xest["x" + (i + 17)]
10 );
11
12 flow.set("lastActiveObserver", active);
13 flow.set("lastActiveXest", xestArr);
14
15 msg.topic = "local1/topic";
16 node.warn({
17   activeObserver: active,
18   observer: msg.observer,
19   xest: msg.payload.xest
20 });
21 return msg;

```

Figure 34. Gate function for active-observer telemetry arbitration.

The supervisory flow receives the selected local estimate through local1/topic and the target input information through Uf/topic. Two lookup/interpolation functions use CSV files generated offline from the observer-bank construction procedure. These files contain the nominal operating points, target-operating-area bounds, steady-state values, and observer identifiers required by the switching logic. The supervisory decision block combines the incoming payloads, identifies the current target operating area, evaluates the switching condition based on measurable-output convergence, and publishes the new active observer identifier to data/topic.

A reset function was included for repeatable validation. It clears the stored observer states and initializes the Node-RED context variables before a new replay sequence is started. This reset action is used only at the beginning of a validation run and is not part of the normal online operation of the soft sensor.

Although the observer-bank construction procedure of Section 4 produced 242 retained observer–area pairs for the selected normalized input domain, the validation trajectory activates only the observer path $8 \rightarrow 2 \rightarrow 13 \rightarrow 1$. It should be noted that, for demonstration purposes only, the Node-RED instance implemented the subset of observers that were activated during the considered validation scenario. Implementing the full previously determined and evaluated observer bank leads to the same results, since the additional observers remain inactive throughout the experiment.

The Node-RED implementation logs the selected estimate, active observer identifier, switching instants, timestamps, and reference values. The logs are exported automatically to CSV files and compared with the MATLAB reference trajectories. The comparison confirms that the Node-RED implementation reproduces the intended observer scheduling, performs the required switching sequence, preserves convergence after transitions, and maintains small estimation errors. Indicatively, in Figures 35–56 the estimations of the chlorine concentrations generated via the Node-RED implementation vs. the respective chlorine concentration generated through the MATLAB simulations in Section 4.2 are presented. Comparing the reference data generated through MATLAB with the corresponding Node-RED estimation data, it can readily be verified that the Node-RED implementation reproduces the reference system dynamics, as well as the reference estimated data with practically identical behavior. The error distribution is strongly concentrated close to

zero. The median absolute error is $0.00098[\text{mg/L}]$; 95% of the absolute errors are below $0.02675[\text{mg/L}]$, and 98.19% are below $0.05[\text{mg/L}]$. The maximum absolute error is $0.12813[\text{mg/L}]$ and occurred for state x_{17} during the initial transient. The results confirm that the Node-RED realization accurately preserves the behavior of the reference implementation.

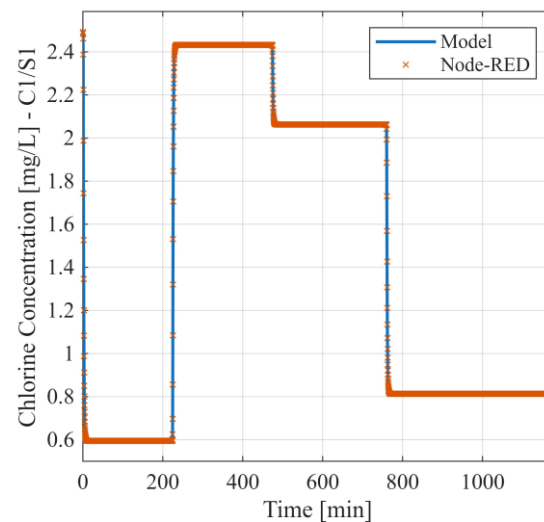


Figure 35. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 1.

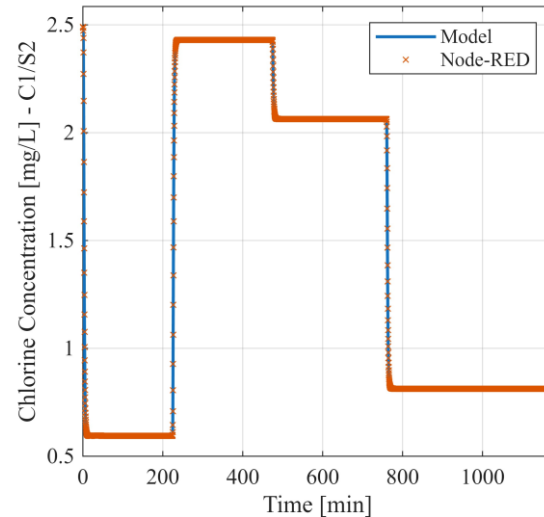


Figure 36. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 1.

It is important to mention that the Node-RED experiment constitutes a simulation-supported edge runtime validation rather than a field trial. The nonlinear MATLAB model represents the water distribution process and generates the telemetry supplied to the Node-RED runtime through MQTT. Consequently, this experiment evaluates the fidelity of the edge implementation, including data acquisition, observer execution, supervisory switching, and telemetry arbitration, relative to the MATLAB reference implementation. It does not independently validate the physical model or the estimation accuracy using measurements obtained from an operating water distribution network. Validation using operational supervisory control and data acquisition (SCADA) data and field chlorine measurements therefore remains a subject of future work.

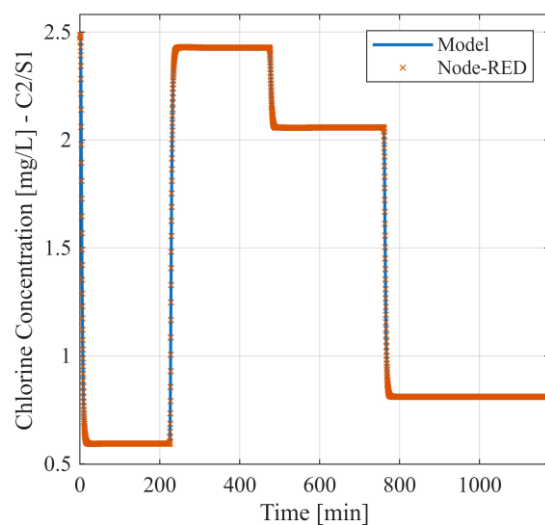


Figure 37. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 2.

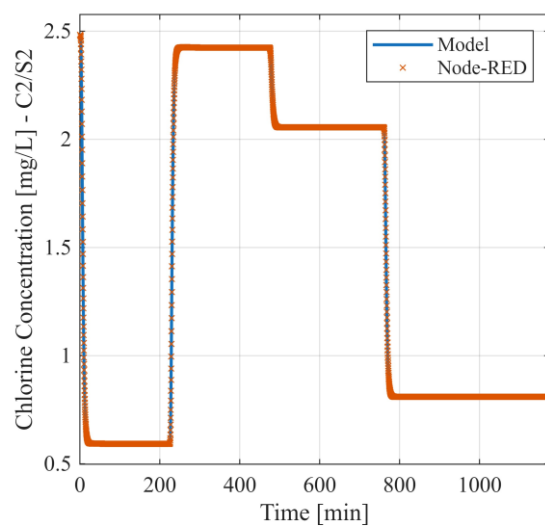


Figure 38. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 2.

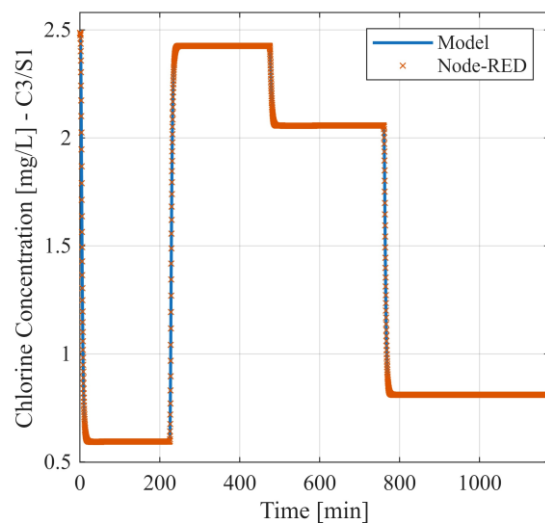


Figure 39. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 3.

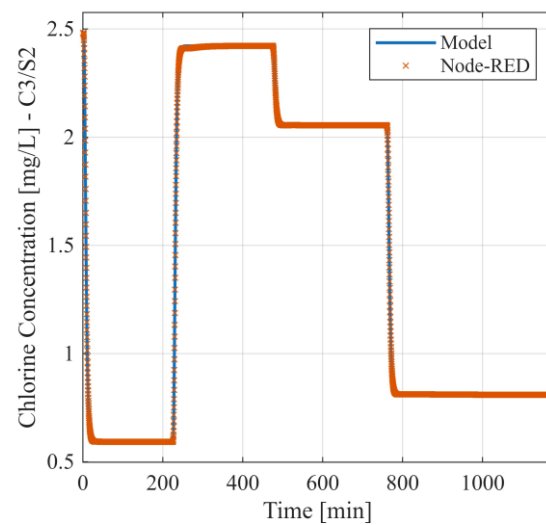


Figure 40. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 3.

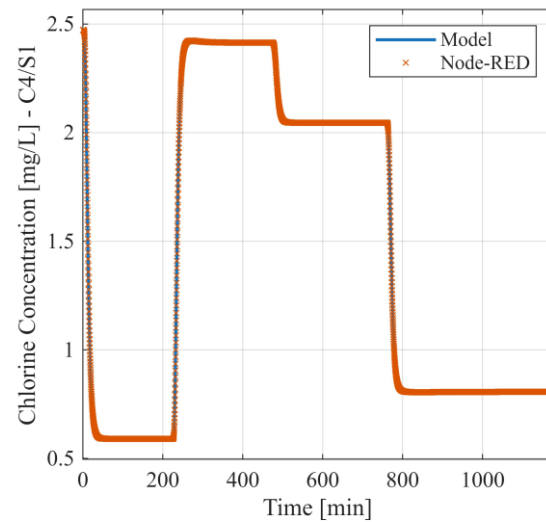


Figure 41. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 4.

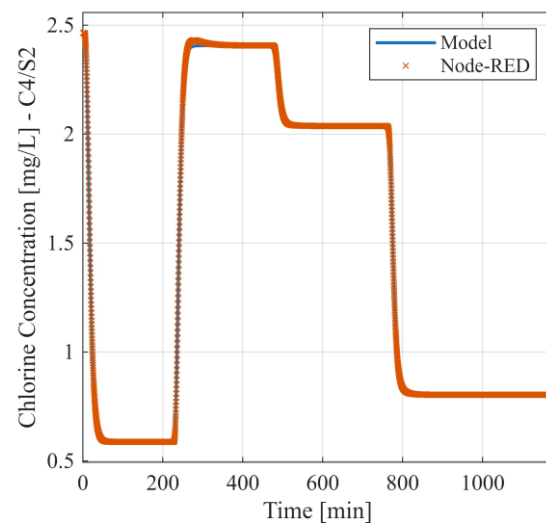


Figure 42. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 4.

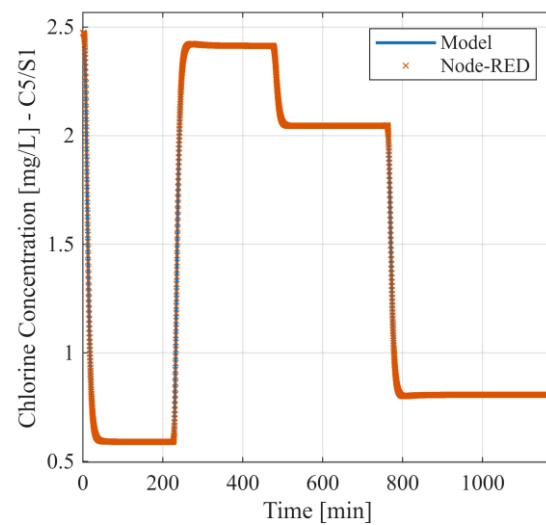


Figure 43. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 5.

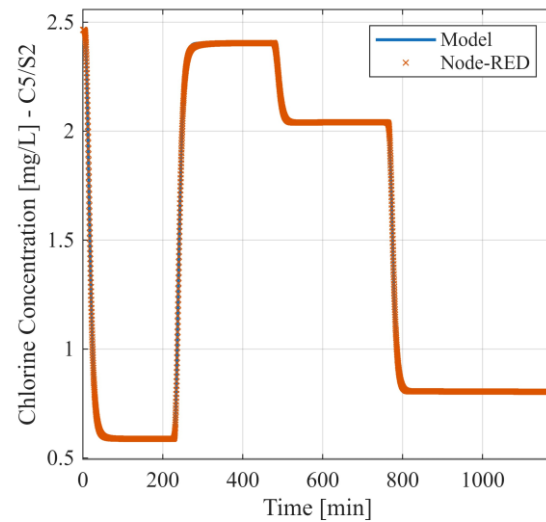


Figure 44. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 5.

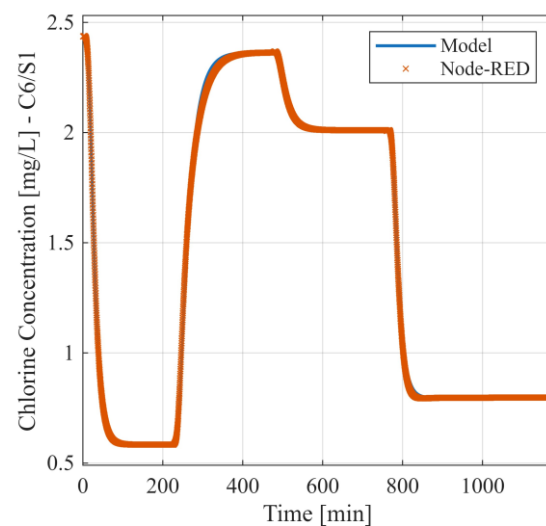


Figure 45. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 6.

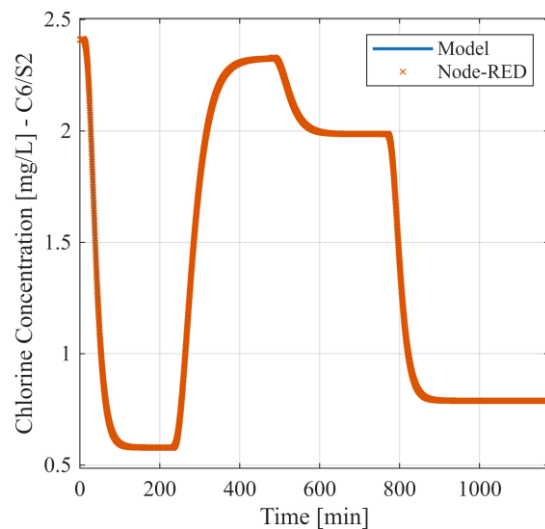


Figure 46. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 6.

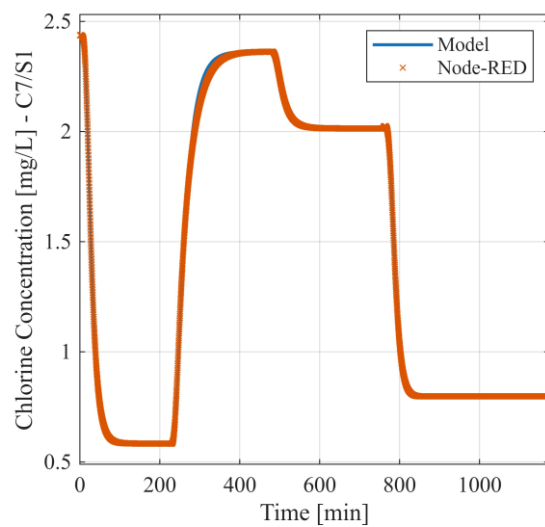


Figure 47. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 7.

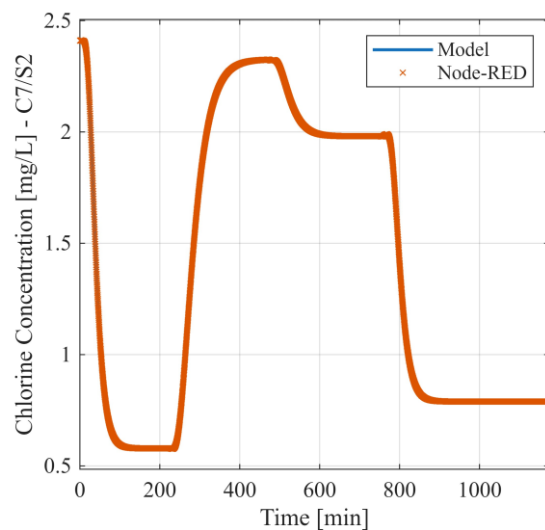


Figure 48. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 7.

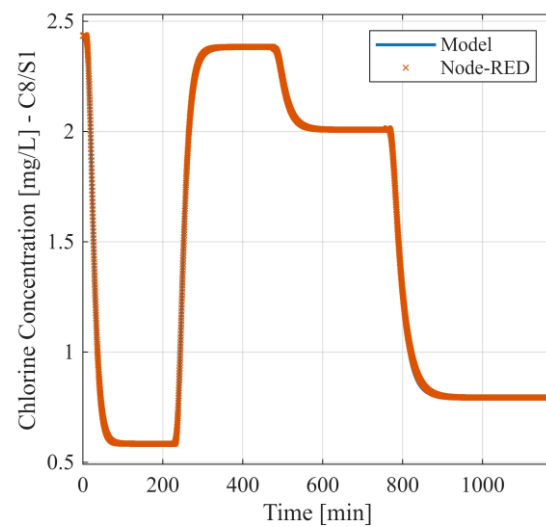


Figure 49. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 8.

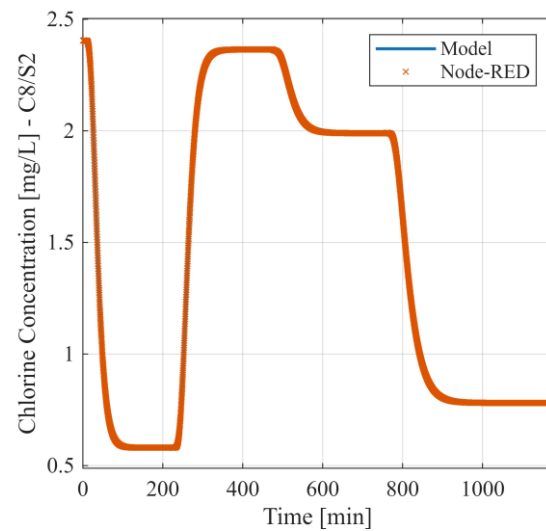


Figure 50. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 8.

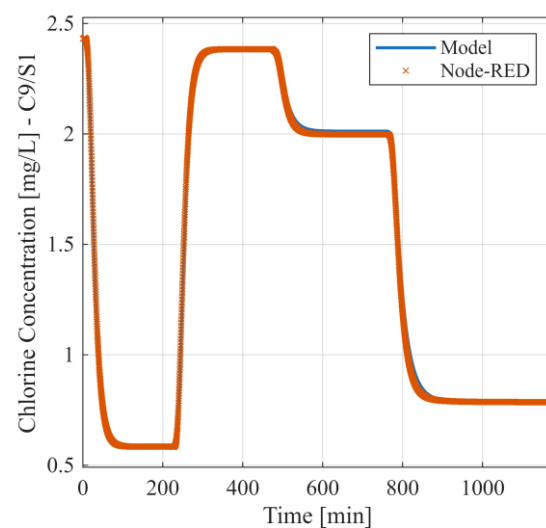


Figure 51. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 9.

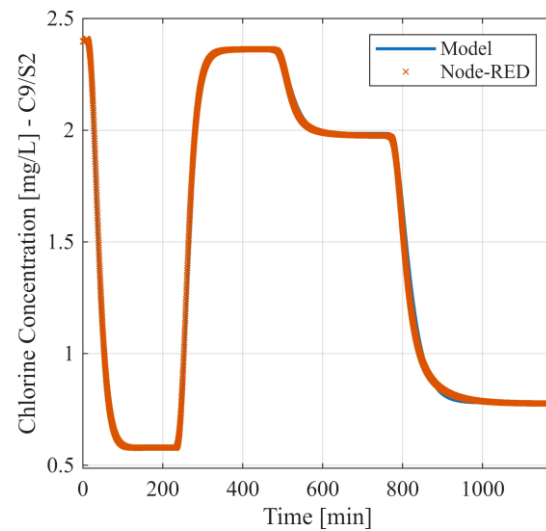


Figure 52. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 9.

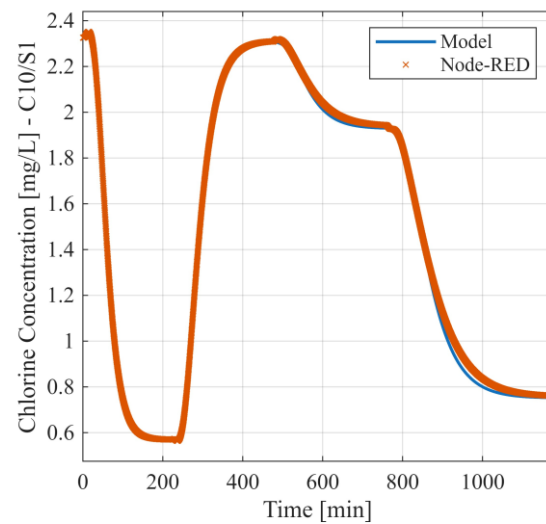


Figure 53. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 10.

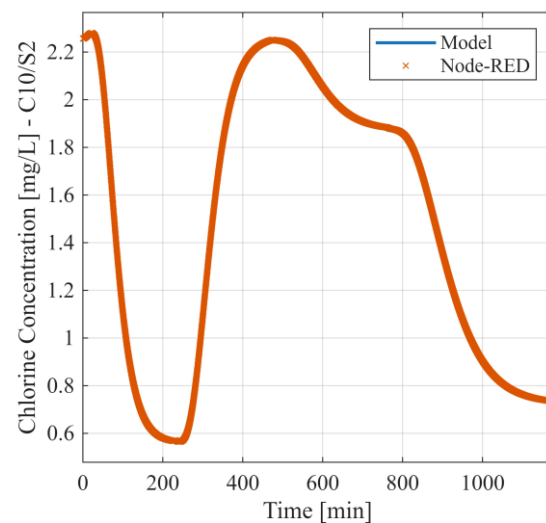


Figure 54. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 10.

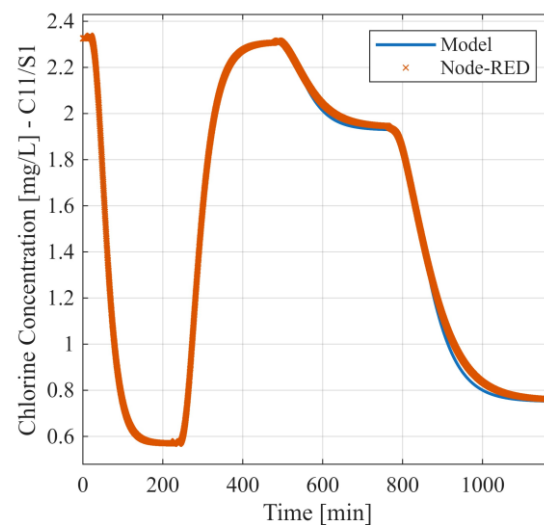


Figure 55. Node-RED chlorine concentration estimation vs. reference in Section 1 of Conduit 11.

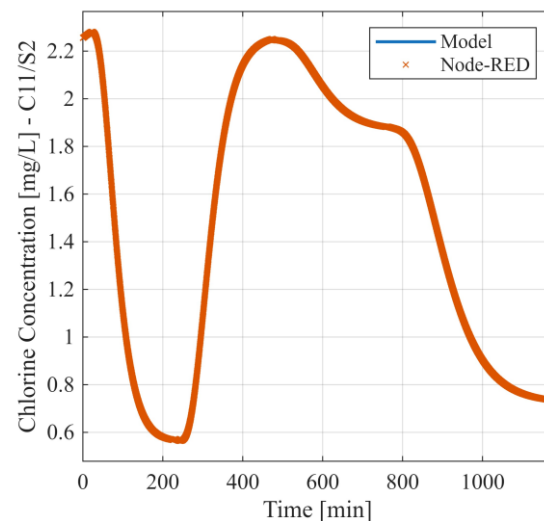


Figure 56. Node-RED chlorine concentration estimation vs. reference in Section 2 of Conduit 11.

6. Conclusions

In the present paper, a reduced observer-bank synthesis and edge-oriented implementation framework for a residual chlorine concentration soft sensor in water distribution networks was presented. The proposed approach was developed for a benchmark network described by nonlinear hydraulic and chlorine transport–reaction dynamics. Around admissible operating points, local discrete-time linear approximants were derived and used as the basis for the design of a finite bank of full-order Luenberger-type observers. The resulting soft sensor estimates nonmeasurable chlorine concentration variables using only a limited set of measurable chlorine outputs, thus addressing the practical limitation of sparse water-quality instrumentation in distribution networks.

A main component of the proposed methodology is the systematic construction of the observer bank. For each admissible operating point, the corresponding nonlinear steady state was computed, a local discrete-time linear approximant was obtained, and the observer parameters were selected through a metaheuristic multicriterion tuning procedure. The tuning combined discrete-time pole-placement constraints with estimation-performance objectives, so that each observer was not only locally stable but also suitable

for chlorine estimation within a prescribed neighborhood of its nominal point. The target operating area of each observer was then determined through validation over sampled operating conditions. A central contribution of the paper was the observer-bank construction algorithm that eliminates redundant target operating areas. Instead of selecting all operating points a priori, the algorithm adaptively covered the normalized input domain by adding new observer–area pairs only where uncovered regions remained. After each new target operating area was generated, the algorithm checked whether any previously retained operating area was fully covered by the union of the remaining ones. Such redundant areas, together with their corresponding observers, were removed without reducing the covered region. In this way, the proposed algorithm produced a compact observer bank, preserving operating-domain coverage while avoiding unnecessary parallel observers and reducing the implementation burden at the edge. The observer bank was combined with a switching supervisor based on target-area overlap and measurable-output convergence. This mechanism enabled the soft sensor to change the active observer during operating transitions while maintaining consistency between the selected observer and the current operating condition. The complete scheme was then transferred to a Node-RED edge runtime. Within this runtime, the observers were executed in parallel, while the supervisor and switching mechanism selected the active estimate and forwarded it through a single telemetry stream. Deterministic measurement data generated by a MATLAB simulation of the benchmark water distribution network were fed to the Node-RED instance, allowing repeatable evaluation of data acquisition, observer execution, switching, telemetry arbitration, and runtime adaptation. The proposed implementation was shown to reconstruct all nonmeasurable chlorine concentrations with high accuracy.

Future research will focus on (a) the development of online adaptation mechanisms, allowing the observer-bank parameters, target operating areas, or switching thresholds to be updated when the hydraulic regime, chlorine reaction parameters, or demand patterns change, and (b) investigation of larger and more complex network topologies, distributed edge deployments and integration with SCADA, IoT, and cloud platforms for operational decision support.

Author Contributions: Conceptualization, N.D.K., F.N.K. and M.P.T.; methodology, N.D.K., F.N.K. and A.N.M.; software, N.D.K., A.N.M. and D.G.F.; validation, N.D.K., A.N.M., D.G.F. and M.P.T.; formal analysis, N.D.K. and F.N.K.; resources, N.D.K., A.N.M. and D.G.F.; data curation, N.D.K. and A.N.M.; writing—original draft preparation, N.D.K., F.N.K. and A.N.M.; writing—review and editing N.D.K., F.N.K. and D.G.F.; visualization, N.D.K., A.N.M. and D.G.F.; supervision, N.D.K. and F.N.K.; project administration., N.D.K. and M.P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: This work was conducted within the “SUB1.1 Clusters of Research Excellence (CREs)” Action of the National Recovery and Resilience Plan “Greece 2.0”, co-funded by Greece and the European Union—NextGenerationEU, under the project “Development of Soft Sensors in Water Distribution Networks (ΥΠ3ΤΑ-0560428-DeSw)”.

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

Appendix A

$$\begin{aligned}
 \gamma_1(\mathbf{x}) &= -\frac{A_1 g x_{12}}{L_1} - \frac{x_1^2 f(x_1, D_1, \varepsilon_1)}{2A_1 D_1}, \quad \gamma_2(\mathbf{x}) = \frac{A_2 g x_{12}}{L_2} - \frac{x_2^2 f(x_2, D_2, \varepsilon_2)}{2A_2 D_2}, \\
 \gamma_3(\mathbf{x}) &= \frac{A_3 g(x_{12} - x_{13})}{L_3} - \frac{x_3^2 f(x_3, D_3, \varepsilon_3)}{2A_3 D_3}, \quad \gamma_4(\mathbf{x}) = \frac{A_4 g(x_{13} - x_{14})}{L_4} - \frac{x_4^2 f(x_4, D_4, \varepsilon_4)}{2A_4 D_4}, \\
 \gamma_5(\mathbf{x}) &= \frac{A_5 g(x_{13} - x_{15})}{L_5} - \frac{x_5^2 f(x_5, D_5, \varepsilon_5)}{2A_5 D_5}, \quad \gamma_6(\mathbf{x}) = \frac{A_6 g x_{14}}{L_6} - \frac{x_6^2 f(x_6, D_6, \varepsilon_6)}{2A_6 D_6}, \\
 \gamma_7(\mathbf{x}) &= \frac{A_7 g x_{14}}{L_7} - \frac{x_7^2 f(x_7, D_7, \varepsilon_7)}{2A_7 D_7}, \quad \gamma_8(\mathbf{x}) = \frac{A_8 g x_{15}}{L_8} - \frac{x_8^2 f(x_8, D_8, \varepsilon_8)}{2A_8 D_8}, \\
 \gamma_9(\mathbf{x}) &= \frac{A_9 g(x_{15} - x_{16})}{L_9} - \frac{x_9^2 f(x_9, D_9, \varepsilon_9)}{2A_9 D_9}, \quad \gamma_{10}(\mathbf{x}) = \frac{A_{10} g x_{16}}{L_{10}} - \frac{x_{10}^2 f(x_{10}, D_{10}, \varepsilon_{10})}{2A_{10} D_{10}}, \\
 \gamma_{11}(\mathbf{x}) &= \frac{A_{11} g x_{16}}{L_{11}} - \frac{x_{11}^2 f(x_{11}, D_{11}, \varepsilon_{11})}{2A_{11} D_{11}}, \quad \gamma_{12}(\mathbf{x}) = \frac{b^2(x_1 - x_2 - x_3)}{A_1 g L_1}, \\
 \gamma_{13}(\mathbf{x}) &= \frac{b^2(x_3 - x_4 - x_5)}{A_3 g L_3}, \quad \gamma_{14}(\mathbf{x}) = \frac{b^2(x_4 - x_6 - x_7)}{A_4 g L_4}, \\
 \gamma_{15}(\mathbf{x}) &= \frac{b^2(x_5 - x_8 - x_9)}{A_5 g L_5}, \quad \gamma_{16}(\mathbf{x}) = \frac{b^2(x_9 - x_{10} - x_{11})}{A_9 g L_9}, \\
 \gamma_{17}(\mathbf{x}) &= \left(-k_b - \frac{4n_{cl} x_1}{D_1^2 L_1 \pi}\right) x_{17}, \quad \gamma_{18}(\mathbf{x}) = \frac{4n_{cl} x_1 (x_{17} - x_{18})}{D_1^2 L_1 \pi} - k_b x_{18}, \\
 \gamma_{19}(\mathbf{x}) &= -k_b x_{19} + \frac{4n_{cl} (x_{18} - x_{19}) x_2}{D_2^2 L_2 \pi}, \quad \gamma_{20}(\mathbf{x}) = \frac{4n_{cl} x_2 (x_{19} - x_{20})}{D_2^2 L_2 \pi} - k_b x_{20}, \\
 \gamma_{21}(\mathbf{x}) &= -k_b x_{21} + \frac{4n_{cl} (x_{18} - x_{21}) x_3}{D_3^2 L_3 \pi}, \quad \gamma_{22}(\mathbf{x}) = -k_b x_{22} + \frac{4n_{cl} (x_{21} - x_{22}) x_3}{D_3^2 L_3 \pi}, \\
 \gamma_{23}(\mathbf{x}) &= -k_b x_{23} + \frac{4n_{cl} (x_{22} - x_{23}) x_4}{D_4^2 L_4 \pi}, \quad \gamma_{24}(\mathbf{x}) = -k_b x_{24} + \frac{4n_{cl} (x_{23} - x_{24}) x_4}{D_4^2 L_4 \pi}, \\
 \gamma_{25}(\mathbf{x}) &= -k_b x_{25} + \frac{4n_{cl} (x_{22} - x_{25}) x_5}{D_5^2 L_5 \pi}, \quad \gamma_{26}(\mathbf{x}) = -k_b x_{26} + \frac{4n_{cl} (x_{25} - x_{26}) x_5}{D_5^2 L_5 \pi}, \\
 \gamma_{27}(\mathbf{x}) &= -k_b x_{27} + \frac{4n_{cl} (x_{24} - x_{27}) x_6}{D_6^2 L_6 \pi}, \quad \gamma_{28}(\mathbf{x}) = -k_b x_{28} + \frac{4n_{cl} (x_{27} - x_{28}) x_6}{D_6^2 L_6 \pi}, \\
 \gamma_{29}(\mathbf{x}) &= -k_b x_{29} + \frac{4n_{cl} (x_{24} - x_{29}) x_7}{D_7^2 L_7 \pi}, \quad \gamma_{30}(\mathbf{x}) = -k_b x_{30} + \frac{4n_{cl} (x_{29} - x_{30}) x_7}{D_7^2 L_7 \pi}, \\
 \gamma_{31}(\mathbf{x}) &= -k_b x_{31} + \frac{4n_{cl} (x_{26} - x_{31}) x_8}{D_8^2 L_8 \pi}, \quad \gamma_{32}(\mathbf{x}) = -k_b x_{32} + \frac{4n_{cl} (x_{31} - x_{32}) x_8}{D_8^2 L_8 \pi}, \\
 \gamma_{33}(\mathbf{x}) &= -k_b x_{33} + \frac{4n_{cl} (x_{26} - x_{33}) x_9}{D_9^2 L_9 \pi}, \quad \gamma_{34}(\mathbf{x}) = -k_b x_{34} + \frac{4n_{cl} (x_{33} - x_{34}) x_9}{D_9^2 L_9 \pi}, \\
 \gamma_{35}(\mathbf{x}) &= \frac{4n_{cl} x_{10} (x_{34} - x_{35})}{D_{10}^2 L_{10} \pi} - k_b x_{35}, \quad \gamma_{36}(\mathbf{x}) = \frac{4n_{cl} x_{10} (x_{35} - x_{36})}{D_{10}^2 L_{10} \pi} - k_b x_{36}, \\
 \gamma_{37}(\mathbf{x}) &= \frac{4n_{cl} x_{11} (x_{34} - x_{37})}{D_{11}^2 L_{11} \pi} - k_b x_{37}, \quad \gamma_{38}(\mathbf{x}) = \frac{4n_{cl} x_{11} (x_{37} - x_{38})}{D_{11}^2 L_{11} \pi} - k_b x_{38}, \\
 z_{1,1}(\mathbf{x}) &= \frac{A_1 g}{L_1}, \quad z_{2,2}(\mathbf{x}) = -\frac{A_2 g}{L_2}, \quad z_{6,3}(\mathbf{x}) = -\frac{A_6 g}{L_6}, \quad z_{7,4}(\mathbf{x}) = -\frac{A_7 g}{L_7}, \quad z_{8,5}(\mathbf{x}) = -\frac{A_8 g}{L_8}, \\
 z_{10,6}(\mathbf{x}) &= -\frac{A_{10} g}{L_{10}}, \quad z_{11,7}(\mathbf{x}) = -\frac{A_{11} g}{L_{11}}, \quad z_{17,8}(\mathbf{x}) = \frac{4n_{cl} x_1}{D_1^2 L_1 \pi}, \quad z_{12,9}(\mathbf{x}) = -\frac{b^2}{A_1 g L_1}, \\
 z_{13,10}(\mathbf{x}) &= -\frac{b^2}{A_3 g L_3}, \quad z_{14,11}(\mathbf{x}) = -\frac{b^2}{A_4 g L_4}, \quad z_{15,12}(\mathbf{x}) = -\frac{b^2}{A_5 g L_5}, \quad z_{16,13}(\mathbf{x}) = -\frac{b^2}{A_9 g L_9},
 \end{aligned}$$

$$f(Q, D, \varepsilon) = \lambda_1 \left[\varepsilon / D + D\pi\lambda_2\mu / (4Q\rho) \right]^{1/4},$$

where L_i and D_i are the length and inner diameter of conduit i , $A_i = \pi D_i^2 / 4$, ε_i is the absolute roughness of conduit i , g is the magnitude of the gravitational acceleration vector, b is the pressure wave speed, ρ and μ are the density and dynamic viscosity of the water, n_{cl} is the number of spatial discretization sections ($n_{cl} = 2$), k_b is a first-order chlorine reaction rate, while λ_1 and λ_2 are positive pipe friction model parameters.

Appendix B

$$\begin{aligned} \bar{u}_1 &= \bar{x}_{12} + \frac{L_1 \bar{x}_1^2 f(\bar{x}_1, D_1, \varepsilon_1)}{2A_1^2 D_1 g}, \quad \bar{u}_2 = \bar{x}_{12} - \frac{L_2 \bar{x}_2^2 f(\bar{x}_2, D_2, \varepsilon_2)}{2A_2^2 D_2 g}, \\ \bar{u}_3 &= \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_4 \bar{x}_4^2 f(\bar{x}_4, D_4, \varepsilon_4)}{2gA_4^2 D_4} + \frac{L_6 \bar{x}_6^2 f(\bar{x}_6, D_6, \varepsilon_6)}{2gA_6^2 D_6} \right), \\ \bar{u}_4 &= \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_4 \bar{x}_4^2 f(\bar{x}_4, D_4, \varepsilon_4)}{2gA_4^2 D_4} + \frac{L_7 \bar{x}_7^2 f(\bar{x}_7, D_7, \varepsilon_7)}{2gA_7^2 D_7} \right), \\ \bar{u}_5 &= \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_5 \bar{x}_5^2 f(\bar{x}_5, D_5, \varepsilon_5)}{2gA_5^2 D_5} + \frac{L_8 \bar{x}_8^2 f(\bar{x}_8, D_8, \varepsilon_8)}{2gA_8^2 D_8} \right), \\ \bar{u}_6 &= \bar{x}_{12} - \left(\frac{L_{10} \bar{x}_{10}^2 f(\bar{x}_{10}, D_{10}, \varepsilon_{10})}{2gA_{10}^2 D_{10}} + \frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_5 \bar{x}_5^2 f(\bar{x}_5, D_5, \varepsilon_5)}{2gA_5^2 D_5} + \frac{L_9 \bar{x}_9^2 f(\bar{x}_9, D_9, \varepsilon_9)}{2gA_9^2 D_9} \right), \\ \bar{u}_7 &= \bar{x}_{12} - \left(\frac{L_{11} \bar{x}_{11}^2 f(\bar{x}_{11}, D_{11}, \varepsilon_{11})}{2g_{11}^2 A_{11}^2 D_{11}} + \frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_5 \bar{x}_5^2 f(\bar{x}_5, D_5, \varepsilon_5)}{2gA_5^2 D_5} + \frac{L_9 \bar{x}_9^2 f(\bar{x}_9, D_9, \varepsilon_9)}{2gA_9^2 D_9} \right), \\ \bar{u}_9 &= \bar{x}_1 - \bar{x}_2 - \bar{x}_3, \quad \bar{u}_{10} = \bar{x}_3 - \bar{x}_4 - \bar{x}_5, \quad \bar{u}_{11} = \bar{x}_4 - \bar{x}_6 - \bar{x}_7, \quad \bar{u}_{12} = \bar{x}_5 - \bar{x}_8 - \bar{x}_9, \quad \bar{u}_{13} = \bar{x}_9 - \bar{x}_{10} - \bar{x}_{11}, \\ \bar{x}_{13} &= \bar{x}_{12} - \frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2A_3^2 D_3 g}, \quad \bar{x}_{14} = \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_4 \bar{x}_4^2 f(\bar{x}_4, D_4, \varepsilon_4)}{2gA_4^2 D_4} \right), \\ \bar{x}_{15} &= \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_5 \bar{x}_5^2 f(\bar{x}_5, D_5, \varepsilon_5)}{2gA_5^2 D_5} \right), \\ \bar{x}_{16} &= \bar{x}_{12} - \left(\frac{L_3 \bar{x}_3^2 f(\bar{x}_3, D_3, \varepsilon_3)}{2gA_3^2 D_3} + \frac{L_5 \bar{x}_5^2 f(\bar{x}_5, D_5, \varepsilon_5)}{2gA_5^2 D_5} + \frac{L_9 \bar{x}_9^2 f(\bar{x}_9, D_9, \varepsilon_9)}{2gA_9^2 D_9} \right), \\ \bar{x}_{17} &= \frac{4n_{cl} \bar{u}_8 \bar{x}_1}{D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1}, \quad \bar{x}_{18} = \frac{16n_{cl}^2 \bar{u}_8 \bar{x}_1^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2}, \\ \bar{x}_{19} &= \frac{64n_{cl}^3 \bar{u}_8 \bar{x}_1^2 \bar{x}_2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_2^2 k_b L_2 \pi + 4n_{cl} \bar{x}_2)}, \quad \bar{x}_{20} = \frac{256n_{cl}^4 \bar{u}_8 \bar{x}_1^2 \bar{x}_2^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_2^2 k_b L_2 \pi + 4n_{cl} \bar{x}_2)^2}, \\ \bar{x}_{21} &= \frac{64n_{cl}^3 \bar{u}_8 \bar{x}_1^2 \bar{x}_3}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)}, \quad \bar{x}_{22} = \frac{256n_{cl}^4 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2}, \\ \bar{x}_{23} &= \frac{1024n_{cl}^5 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)}, \\ \bar{x}_{24} &= \frac{4096n_{cl}^6 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)^2}, \\ \bar{x}_{25} &= \frac{1024n_{cl}^5 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)} \end{aligned}$$

$$\begin{aligned}
\bar{x}_{26} &= \frac{4096n_{cl}^6 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2} \\
\bar{x}_{27} &= \frac{16384n_{cl}^7 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4^2 \bar{x}_6}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)^2 (D_6^2 k_b L_6 \pi + 4n_{cl} \bar{x}_6)^2} \\
\bar{x}_{28} &= \frac{65536n_{cl}^8 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4^2 \bar{x}_6^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)^2 (D_6^2 k_b L_6 \pi + 4n_{cl} \bar{x}_6)^2} \\
\bar{x}_{29} &= \frac{16384n_{cl}^7 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4^2 \bar{x}_7}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)^2 (D_7^2 k_b L_7 \pi + 4n_{cl} \bar{x}_7)^2} \\
\bar{x}_{30} &= \frac{65536n_{cl}^8 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_4^2 \bar{x}_7^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_4^2 k_b L_4 \pi + 4n_{cl} \bar{x}_4)^2 (D_7^2 k_b L_7 \pi + 4n_{cl} \bar{x}_7)^2} \\
\bar{x}_{31} &= \frac{16384n_{cl}^7 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_8}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_8^2 k_b L_8 \pi + 4n_{cl} \bar{x}_8)^2} \\
\bar{x}_{32} &= \frac{65536n_{cl}^8 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_8^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_8^2 k_b L_8 \pi + 4n_{cl} \bar{x}_8)^2} \\
\bar{x}_{33} &= \frac{16384n_{cl}^7 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2} \\
\bar{x}_{34} &= \frac{65536n_{cl}^8 \bar{u}_8 \bar{x}_1^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2} \\
\bar{x}_{35} &= \frac{262144n_{cl}^9 \bar{u}_8 \bar{x}_1^2 \bar{x}_{10}^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_{10}^2 k_b L_{10} \pi + 4n_{cl} \bar{x}_{10})^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2} \\
\bar{x}_{36} &= \frac{1048576n_{cl}^{10} \bar{u}_8 \bar{x}_1^2 \bar{x}_{10}^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_{10}^2 k_b L_{10} \pi + 4n_{cl} \bar{x}_{10})^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2} \\
\bar{x}_{37} &= \frac{262144n_{cl}^9 \bar{u}_8 \bar{x}_1^2 \bar{x}_{11}^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_{11}^2 k_b L_{11} \pi + 4n_{cl} \bar{x}_{11})^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2} \\
\bar{x}_{38} &= \frac{1048576n_{cl}^{10} \bar{u}_8 \bar{x}_1^2 \bar{x}_{11}^2 \bar{x}_3^2 \bar{x}_5^2 \bar{x}_9^2}{(D_1^2 k_b L_1 \pi + 4n_{cl} \bar{x}_1)^2 (D_{11}^2 k_b L_{11} \pi + 4n_{cl} \bar{x}_{11})^2 (D_3^2 k_b L_3 \pi + 4n_{cl} \bar{x}_3)^2 (D_5^2 k_b L_5 \pi + 4n_{cl} \bar{x}_5)^2 (D_9^2 k_b L_9 \pi + 4n_{cl} \bar{x}_9)^2}
\end{aligned}$$

References

- Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. <https://doi.org/10.1016/j.compchemeng.2008.12.012>.
- Koumboulis, F.N.; Fragkoulis, D.G.; Kouvakas, N.D.; Feidopiasti, A. Soft sensor design via switching observers. *Sensors* **2023**, *23*, 2114. <https://doi.org/10.3390/s23042114>.
- Kouvakas, N.D.; Koumboulis, F.N.; Tzamtzi, M.P.; Fragkoulis, D.G. Toward the design of chlorine soft sensors via stepwise safe switching observers for a primary water distribution network. In *Artificial Intelligence Applications in Engineering and Medicine: Proceedings of the 2nd Olympiad in Engineering Science*; Pavlou, D., Adeli, H., Correia, J., Eds.; Structural Integrity; Springer: Cham, Switzerland, 2026; Volume 36.
- Kouvakas, N.D.; Koumboulis, F.N.; Tzamtzi, M.P.; Fragkoulis, D.G.; Katsiavrias, K.K.; Katsiavrias, K.; Dimitropoulos, J. On the Discrete Time Stepwise Safe Switching Soft Sensor Design for Water Distribution Networks. In *Proceedings of the 12th International Conference on Control Engineering & Information Technology (CEIT 2026)*, Istanbul, Türkiye, 27–30 April 2026; Volume 98, pp. 14–20.
- Koumboulis, F.N.; Kouvakas, N.D.; Tzamtzi, M.P.; Fragkoulis, D.G.; Menexis, A.N.; Katsiavrias, K.K.; Katsiavrias, K.; Dimitropoulos, J. Toward Edge Implementation of Soft Sensors for Water Distribution Networks. In *Proceedings of the 12th*

- International Conference on Control Engineering & Information Technology (CEIT 2026), Istanbul, Türkiye, 27–30 April 2026; Volume 98, pp. 28–34.
6. Chebbi, A.; Franchek, M.A.; Grigoriadis, K. Simultaneous state and parameter estimation methods based on Kalman filters and Luenberger observers: A tutorial & review. *Sensors* **2025**, *25*, 7043. <https://doi.org/10.3390/s25227043>.
 7. Xu, F.; Wang, J.; Sakurai, K.; Sakai, Y.; Sabu, S.; Kanayama, H.; Zhang, R.; Satou, D.; Kansha, Y. Soft-sensor model for indoor temperature prediction under heating conditions. *Therm. Sci. Eng. Prog.* **2024**, *51*, 102650. <https://doi.org/10.1016/j.tsep.2024.102650>.
 8. Nascimento, A.S.B.S.; Machado, J.P.Z.; Coelho, L.S.; Flesch, R.C.C. Evaporating temperature estimation of refrigeration systems based on vibration data-driven soft sensors. *Int. J. Refrig.* **2024**, *168*, 288–296. <https://doi.org/10.1016/j.ijrefrig.2024.08.020>.
 9. Yan, M.; Bi, H.; Wang, H.; Xu, C.; Chen, L.; Zhang, L.; Chen, S.; Xu, X.; Chen, Q.; Jia, Y.; et al. Advanced soft-sensing techniques for predicting furnace temperature in industrial organic waste gasification. *Process Saf. Environ. Prot.* **2024**, *190*, 1253–1262. <https://doi.org/10.1016/j.psep.2024.07.124>.
 10. He, Z.; Wang, Y.; Yang, G.; Han, C.; Gao, J.; Xu, S.; Yin, G.; Tian, X.; Wang, Z.; Peng, X. Data-driven high-temperature super-heater wall temperature prediction using Polar Lights optimized Kolmogorov–Arnold networks. *Processes* **2025**, *13*, 3741. <https://doi.org/10.3390/pr13113741>.
 11. Junge, T.; Mehner, T.; Nestler, A.; Lampke, T.; Schubert, A. Methodology for soft-sensor design and in-process surface conditioning in turning of aluminum alloys. *Prod. Eng. Res. Devel.* **2024**, *18*, 267–287. <https://doi.org/10.1007/s11740-023-01260-0>.
 12. Böttger, D.; González, G.; Geiser, A.; Kempf, D.; Lanza, G.; Schulze, V.; Wolter, B. Soft sensor for in-line quality control of turning processes based on non-destructive testing techniques and advanced data fusion. *Prod. Eng. Res. Devel.* **2024**, *18*, 197–206. <https://doi.org/10.1007/s11740-023-01254-y>.
 13. Liu, Z.; Liu, H.; Chen, F.; Li, H.; Xue, X. Dynamic soft sensor model for endpoint carbon content and temperature in BOF steelmaking based on adaptive feature matching variational autoencoder. *Processes* **2024**, *12*, 1807. <https://doi.org/10.3390/pr12091807>.
 14. Yang, Y.; Zhou, Y.; Yao, D.; Liu, X. Mechanism-constrained multistage recursive soft sensor framework for slab temperature prediction in reheating furnace. *IEEE Sens. J.* **2025**, *25*, 20970–20981. <https://doi.org/10.1109/JSEN.2025.3563586>.
 15. Li, D.; Yang, C.; Li, Y. A multi-subsystem collaborative Bi-LSTM-based adaptive soft sensor for global prediction of ammonia-nitrogen concentration in wastewater treatment processes. *Water Res.* **2024**, *254*, 121347. <https://doi.org/10.1016/j.watres.2024.121347>.
 16. Liu, L.; Tian, X.; Ma, Y.; Lu, W.; Luo, Y. Online soft measurement method for chemical oxygen demand based on CNN-BiLSTM-Attention algorithm. *PLoS ONE* **2024**, *19*, e0305216. <https://doi.org/10.1371/journal.pone.0305216>.
 17. Li, W.; Li, Y.; Li, D.; Zhou, J. A multivariable probability density-based auto-reconstruction Bi-LSTM soft sensor for predicting effluent BOD in wastewater treatment plants. *Sensors* **2024**, *24*, 7508. <https://doi.org/10.3390/s24237508>.
 18. Voipan, D.; Voipan, A.E.; Barbu, M. Evaluating machine learning-based soft sensors for effluent quality prediction in wastewater treatment under variable weather conditions. *Sensors* **2025**, *25*, 1692. <https://doi.org/10.3390/s25061692>.
 19. Wang, B.; Yu, A.; Wang, H.; Liu, J. Modeling and optimization of an enhanced soft sensor for the fermentation process of *Pichia pastoris*. *Sensors* **2024**, *24*, 3017. <https://doi.org/10.3390/s24103017>.
 20. Delgado, E.; Moreno, J.C.; Rodríguez-Miranda, E.; Baños, A.; Barreiro, A.; Guzmán, J.L. Soft-sensor based on sliding modes for industrial raceway photobioreactors. *Biosyst. Eng.* **2024**, *246*, 1–12. <https://doi.org/10.1016/j.biosystemseng.2024.07.015>.
 21. Huang, F.; Li, L.; Du, C.; Wang, S.; Liu, X. Soft sensing modeling of penicillin fermentation process based on local selection ensemble learning. *Sci. Rep.* **2024**, *14*, 20349. <https://doi.org/10.1038/s41598-024-71161-4>.
 22. Cheng, X.; Yu, Z.; Wang, G.; Jiang, Q.; Cao, Z. Semi-supervised soft sensor method for fermentation processes based on physical monotonicity and variational autoencoders. *Eng. Appl. Artif. Intell.* **2024**, *137*, 109065. <https://doi.org/10.1016/j.engappai.2024.109065>.
 23. Kouvakas, N.D.; Koumboulis, F.N.; Fragkoulis, D.G.; Fragulis, G.F. Metaheuristic procedures for the determination of a bank of switching observers toward soft sensor design with application to an alcoholic fermentation process. *Mathematics* **2023**, *11*, 4733. <https://doi.org/10.3390/math11234733>.
 24. Bartos, M.; Thomas, M.; Kim, M.-G.; Frankel, M.; Sela, L. Online state estimation in water distribution systems via extended Kalman filtering. *Water Res.* **2024**, *264*, 122201. <https://doi.org/10.1016/j.watres.2024.122201>.
 25. Torres, L.; Jiménez-Cabas, J.; Ponsart, J.-C.; Theilliol, D.; Jiménez-Magaña, M.R.; Guzmán, J.E.V. Pipe roughness calibration approach for water distribution network models using a nonlinear state observer. *Results Eng.* **2024**, *23*, 102713. <https://doi.org/10.1016/j.rineng.2024.102713>.

26. Vrachimis, S.G.; Eliades, D.G.; Polycarpou, M.M. Real-time hydraulic interval state estimation for water transport networks: A case study. *Drink. Water Eng. Sci.* **2018**, *11*, 19–24. <https://doi.org/10.5194/dwes-11-19-2018>.
27. López-Estrada, F.-R.; Gómez-Coronel, L.; Torres, L.; Valencia-Palomo, G.; Santos-Ruiz, I.; Cano, A. Online nodal demand estimation in branched water distribution systems using an array of extended Kalman filters. *Sensors* **2025**, *25*, 2632. <https://doi.org/10.3390/s25082632>.
28. Navarro-Díaz, A.; Delgado-Aguíñaga, J.A.; Santos-Ruiz, I.; Puig, V. Real-time leak diagnosis in water distribution systems based on a bank of observers and a genetic algorithm. *Water* **2022**, *14*, 3289. <https://doi.org/10.3390/w14203289>.
29. Djerioui, M.; Bouamar, M.; Ladjal, M.; Zerguine, A. Chlorine soft sensor based on extreme learning machine for water quality monitoring. *Arab. J. Sci. Eng.* **2019**, *44*, 2033–2044. <https://doi.org/10.1007/s13369-018-3253-8>.
30. Łangowski, R.; Brdyś, M. Monitoring of chlorine concentration in drinking water distribution systems using an interval estimator. *Int. J. Appl. Math. Comput. Sci.* **2007**, *17*, 199–216. <https://doi.org/10.2478/v10006-007-0019-y>.
31. Rajakumar, A.G.; Mohan Kumar, M.S.; Amrutur, B.; Kapelan, Z. Real-time water quality modeling with ensemble Kalman filter for state and parameter estimation in water distribution networks. *J. Water Resour. Plan. Manag.* **2019**, *145*, 04019049. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001118](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001118).
32. Nair, A.M.; Hykkerud, A.; Ratnaweera, H. A cost-effective IoT strategy for remote deployment of soft sensors: A case study on implementing a soft sensor in a multistage MBBR plant. *Water Sci. Technol.* **2020**, *81*, 1733–1739. <https://doi.org/10.2166/wst.2020.067>.
33. Nair, A.; Hykkerud, A.; Ratnaweera, H. Estimating phosphorus and COD concentrations using a hybrid soft sensor: A case study in a Norwegian municipal wastewater treatment plant. *Water* **2022**, *14*, 332. <https://doi.org/10.3390/w14030332>.
34. Pattnaik, B.S.; Pattanayak, A.S.; Udgata, S.K.; Panda, A.K. Machine learning based soft sensor model for BOD estimation using intelligence at edge. *Complex Intell. Syst.* **2021**, *7*, 961–976. <https://doi.org/10.1007/s40747-020-00259-9>.
35. Chaves, A.J.; Martín, C.; Llopis Torres, L.; Díaz, M.; Fernández-Ortega, J.; Barberá, J.A.; Andreo, B. A soft sensor open-source methodology for inexpensive monitoring of water quality: A case study of NO₃[−] concentrations. *J. Comput. Sci.* **2025**, *85*, 102522. <https://doi.org/10.1016/j.jocs.2024.102522>.
36. Ling, T.; Qian, C.; Schiele, G. On-device soft sensors: Real-time fluid flow estimation from level sensor data. In *Mobile and Ubiquitous Systems: Computing, Networking and Services*; Zaslavsky, A., Ning, Z., Kalogeraki, V., Georgakopoulos, D., Chrysanthis, P.K., Eds.; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer: Cham, Switzerland, 2024; Volume 594, pp. 529–537. https://doi.org/10.1007/978-3-031-63992-0_36.
37. Ling, T.; Hoefer, J.; Qian, C.; Schiele, G. FlowPrecision: Advancing FPGA-based real-time fluid flow estimation with linear quantization. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Biarritz, France, 11–15 March 2024; pp. 733–738. <https://doi.org/10.1109/PerComWorkshops59983.2024.10503436>.
38. Ling, T.; Qian, C.; Schiele, G. Towards Auto-Building of Embedded FPGA-Based Soft Sensors for Wastewater Flow Estimation. In Proceedings of the 2024 IEEE Annual Congress on Artificial Intelligence of Things (AIoT), Melbourne, Australia, 24–26 July 2024; IEEE Document 10677629.
39. Ling, T.; Qian, C.; Klann, T.M.; Hoefer, J.; Einhaus, L.; Schiele, G. Configurable multi-layer perceptron-based soft sensors on embedded field programmable gate arrays: Targeting diverse deployment goals in fluid flow estimation. *Sensors* **2025**, *25*, 83. <https://doi.org/10.3390/s25010083>.
40. Vijaya Krishna, V.V.S.; Pappa, N.; Joy Vasantharani, S.P. Realization of deep learning based embedded soft sensor for bioprocess application. *Intell. Autom. Soft Comput.* **2022**, *32*, 781–794. <https://doi.org/10.32604/iasc.2022.022181>.
41. Vijaya Krishna, V.S. Embedded soft sensor for bioreactor application using soft processor of Zynq SoC. *Int. J. Perform. Eng.* **2024**, *20*, 563–571. <https://doi.org/10.23940/ijpe.24.09.p4.563571>.
42. Caponetto, R.; Dongola, G.; Gallo, A.; Xibilia, M.G. FPGA-based soft sensor for the estimation of the kerosene freezing point. In Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES), Lausanne, Switzerland, 8–10 July 2009; pp. 228–236. <https://doi.org/10.1109/SIES.2009.5196219>.
43. Andrade, P.; Silva, I.; Silva, M.; Flores, T.; Cassiano, J.; Costa, D.G. A TinyML soft-sensor approach for low-cost detection and monitoring of vehicular emissions. *Sensors* **2022**, *22*, 3838. <https://doi.org/10.3390/s22103838>.
44. Wang, Y.; Zheng, X.; Wang, L.; Lu, G.; Jia, Y.; Qiu, Y.; Li, M. Edge-computing based soft sensors with local finite impulse response models for vehicle wheel center loads estimation under multiple working conditions. *Control Eng. Pract.* **2023**, *133*, 105447. <https://doi.org/10.1016/j.conengprac.2023.105447>.

45. Moore, B.C. On the flexibility offered by state feedback in multivariable systems beyond closed-loop eigenvalue assignment. *IEEE Trans. Autom. Control* **1976**, *21*, 689–692. <https://doi.org/10.1109/TAC.1976.1101355>.
46. Fahmy, M.M.; O'Reilly, J. Eigenstructure assignment in linear multivariable systems—A parametric solution. *IEEE Trans. Autom. Control* **1983**, *28*, 990–994. <https://doi.org/10.1109/TAC.1983.1103160>.
47. Duan, G.-R. Solutions of the equation $AV + BW = VF$ and their application to eigenstructure assignment in linear systems. *IEEE Trans. Autom. Control* **1993**, *38*, 276–280. <https://doi.org/10.1109/9.250470>.
48. Duan, G.-R.; Wu, Z.-Y.; Howe, D. Explicit parametric solution to observer-based control of self-sensing magnetic bearings. *IFAC Proc.* **1999**, *32*, 7487–7492. [https://doi.org/10.1016/S1474-6670\(17\)57278-X](https://doi.org/10.1016/S1474-6670(17)57278-X).
49. Drosou, T.C.; Kouvakas, N.D.; Koumboulis, F.N.; Tzamtzi, M.P. A mixed analytic/metaheuristic dual stage control scheme toward I/O decoupling for a differential drive mobile robot. In *Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications*; Farmanbar, M., Tzamtzi, M., Verma, A.K., Chakravorty, A., Eds.; Springer: Singapore, 2024; pp. 197–214. https://doi.org/10.1007/978-981-99-9836-4_16.
50. Kouvakas, N.D.; Koumboulis, F.N.; Sigalas, J. A two stage nonlinear I/O decoupling and partially wireless controller for differential drive mobile robots. *Robotics* **2024**, *13*, 26. <https://doi.org/10.3390/robotics13020026>.
51. Bucklew, J.A.; Sethares, W.A. The covering problem and μ -dependent adaptive algorithms. *IEEE Trans. Signal Process.* **1994**, *42*, 2616–2627. <https://doi.org/10.1109/78.324728>.
52. Clarkson, K.L.; Varadarajan, K. Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom.* **2007**, *37*, 43–58. <https://doi.org/10.1007/s00454-006-1273-8>.
53. Overmars, M.H.; Yap, C.-K. New upper bounds in Klee's measure problem. *SIAM J. Comput.* **1991**, *20*, 1034–1045. <https://doi.org/10.1137/0220065>.
54. Chan, T.M. Faster algorithms for largest empty rectangles and boxes. *Discrete Comput. Geom.* **2023**, *70*, 355–375. <https://doi.org/10.1007/s00454-022-00473-x>.

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.