STELIOS VRACHIMIS DOCTOR OF PHILOSOPHY DISSERTATION 2020

UNIVERSITY OF CYPRUS

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING

REAL-TIME STATE ESTIMATION AND FAULT DIAGNOSIS OF WATER DISTRIBUTION SYSTEMS

DOCTOR OF PHILOSOPHY DISSERTATION

STELIOS VRACHIMIS

2020

Πανεπιστήμιο Κύπρου
University of Cyprus

Department of Electrical and Computer Engineering

Real-time State Estimation and Fault Diagnosis of Water Distribution Systems

Stelios G. Vrachimis

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the University of Cyprus

July, 2020

VALIDATION PAGE

Stelios G. Vrachimis

Real-time State Estimation and Fault Diagnosis

of Water Distribution Systems

The present Doctorate Dissertation was submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering, and was approved on July, 9, 2020 by the members of the Examination Committee.

Committee Chair ______________________________

Dr. Christos G. Panayiotou, Professor

Research Supervisor ______________________________

Dr. Marios M. Polycarpou, Professor

Committee Member ______________________________

Dr. Stelios Timotheou, Assistant Professor

Committee Member ______________________________

Dr. Avi Ostfeld, Professor

Committee Member ______________________________

Dr. Symeon Christodoulou, Professor

Committee Member ______________________________

Dr. Demetrios G. Eliades, Research Assistant Professor

DECLARATION OF DOCTORAL CANDIDATE

The present Doctorate Dissertation was submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy of the University of Cyprus. It is a product of original work of my own, unless otherwise mentioned through references, notes, or any other statements.

....

Stelios G. Vrachimis

To my parents, Chrystalla and George

Abstract

Modern societies are starting to grow more conscious of the importance of drinking water, due to water scarcity in many parts of the world as an effect of an increasing human population and climate change. Authorities in many countries are more eager to invest in the maintenance and modernization of Water Distribution Systems (WDS). As a result of aging infrastructures water utilities face problems which were ignored in the past such as high rates of water loss and insufficient monitoring of water quality.

The water industry is being modernized with the installation of sensors for monitoring and control of WDS and computer systems to process these data. State estimation algorithms infer the state of the system, such as water flows and pressures in pipes, tank levels and chlorine concentration in water, using the available measurement-set and system model. A complete view of the distribution network state in real-time supports the decision-making process and enables the efficient operation of these systems, improves customer service and enables the early detection and response to emergency events such as pipe failures or water contamination, thus minimizing their impact. However, state estimation in WDS is a challenging task, mainly due to measurement scarcity which makes large-scale water systems unobservable. To obtain hydraulic observability, pseudo-measurements are often used which are highly uncertain and lack statistical characterization. Water-quality is intrinsically connected with hydraulic dynamics, a fact that makes water-quality models time-varying. Additionally, modeling and parameter uncertainty further increase the difficulty in accurately characterizing water-quality in WDS.

This thesis presents methodologies for state estimation in WDS that address the above practical challenges. The proposed methodologies use available *a priori* information about these systems to derive reliable state estimates which can be utilized by water utilities and practitioners. Specifically, we propose a hydraulic-state estimation approach which assumes bounded measurement and modeling uncertainties and calculates an interval-state estimate considering their combined effect. The reasoning behind this approach is that providing a

range of possible values for each state depending on the level of modeling uncertainty, is often more useful to an operator than providing estimates with no information about the estimation error. The availability of hydraulic-state estimates enables water-quality estimation, due to the interconnected nature of hydraulic and water-quality dynamics. A water-quality state estimation approach is presented which calculates bounds on chlorine concentration and is able to incorporate real-time parameter estimation, considering the existence of unknown water-quality parameters.

The main aim of the proposed practical state estimation approaches is to enable the development of methodologies for fault diagnosis in WDS. Obtaining reliable bounds on state estimates in real-time is a key component of many methodologies related to model-based fault diagnosis. Specifically, we target two type of critical faults in WDS: water leakages and water contamination. Two methodologies for detecting and localizing leakages based on hydraulic interval-state estimation are proposed and evaluated using an open access leakage diagnosis benchmark and established benchmark networks. Water contamination detection is achieved by combining the proposed hydraulic and water-quality state estimation approaches with a detection logic using multi-level thresholds. Moreover, we explore the concept of active fault detection by developing a novel methodology for enhancing the area monitored by water-quality sensors, to address the problem of measurement scarcity.

Περίληψη

Οι σύγχρονες κοινωνίες αρχίζουν να συνειδητοποιούν τη σημασία του πόσιμου νερού λόγω της λειψυδρίας σε πολλά μέρη του κόσμου ως αποτέλεσμα του αυξανόμενου ανθρώπινου πληθυσμού και της κλιματικής αλλαγής. Οι αρχές σε πολλές χώρες είναι πιο πρόθυμες να επενδύσουν στη συντήρηση και τον εκσυγχρονισμό των Συστημάτων Διανομής Νερού (ΣΔΝ). Ως αποτέλεσμα των γερασμένων υποδομών, οι επιχειρήσεις κοινής ωφέλειας αντιμετωπίζουν προβλήματα που αγνοήθηκαν στο παρελθόν, όπως είναι τα υψηλά ποσοστά απώλειας νερού και η ανεπαρκής παρακολούθηση της ποιότητας των υδάτων.

Η βιομηχανία ύδρευσης εκσυγχρονίζεται με την εγκατάσταση αισθητήρων για την παρακολούθηση και τον έλεγχο των ΣΔΝ και υπολογιστικών συστημάτων για την επεξεργασία αυτών των δεδομένων. Οι αλγόριθμοι εκτίμησης κατάστασης συμπεραίνουν την κατάσταση του συστήματος, όπως τις ροές νερού και τις πιέσεις στους αγωγούς, τα επίπεδα των υδατοδεξαμενών και την συγκέντρωση χλωρίου στο νερό, χρησιμοποιώντας τις διαθέσιμες μετρήσεις και μοντέλα του συστήματος. Η πλήρης προβολή της κατάστασης του δικτύου διανομής σε πραγματικό χρόνο υποστηρίζει τη διαδικασία λήψης αποφάσεων και καθιστά δυνατή την αποτελεσματική λειτουργία αυτών των συστημάτων, βελτιώνει την εξυπηρέτηση των πελατών και καθιστά δυνατή την έγκαιρη ανίχνευση και ανταπόκριση σε επείγουσες καταστάσεις όπως σε σπασίματα σωλήνων ή μόλυνση του νερού, ελαχιστοποιώντας έτσι τις επιπτώσεις τους. Ωστόσο, η εκτίμηση της κατάστασης στα ΣΔΝ είναι ένα δύσκολο έργο, κυρίως λόγω της έλλειψης μετρήσεων, που καθιστά τα μεγάλης κλίμακας συστήματα ύδρευσης μη παρατηρήσιμα. Για την επίτευξη υδραυλικής παρατηρησιμότητας γίνεται χρήση ψευδομετρήσεων, οι οποίες είναι εξαιρετικά αβέβαιες και δεν συνοδεύονται με στατιστικό χαρακτηρισμό του σφάλματός τους. Η ποιότητα του νερού είναι εγγενώς συνδεδεμένη με την υδραυλική δυναμική, γεγονός που καθιστά τα μοντέλα ποιότητας νερού χρονομεταβλητά. Επιπλέον, η αβεβαιότητα στα μοντέλα και στις παραμέτρους ποιότητας νερού δυσκολεύουν περαιτέρω τον ακριβή χαρακτηρισμό της ποιότητας στα ΣΔΝ.

Αυτή η διατριβή παρουσιάζει μεθοδολογίες για την εκτίμηση της κατάστασης των ΣΔΝ,

οι οποίες απευθύνουν τις παραπάνω πρακτικές προκλήσεις. Οι προτεινόμενες μεθοδολογίες χρησιμοποιούν διαθέσιμες πληροφορίες για τα συστήματα αυτά αποσκοπώντας στην εξαγωγή αξιόπιστων εκτιμήσεων κατάστασης, οι οποίες μπορούν να χρησιμοποιηθούν από εταιρείες παροχής νερού και επαγγελματίες. Συγκεκριμένα, προτείνουμε μια μέθοδο εκτίμησης υδραυλικής κατάστασης η οποία προϋποθέτει οριοθετημένες αβεβαιότητες στις μετρήσεις και στα μοντέλα και εκτιμά ένα εύρος τιμών για την κατάσταση του συστήματος λαμβάνοντας υπόψη τη συνδυασμένη επίδρασή τους. Η λογική πίσω από αυτή την προσέγγιση είναι ότι η παροχή εύρους πιθανών τιμών για κάθε κατάσταση ανάλογα με το επίπεδο της αβεβαιότητας της μοντελοποίησης, είναι συχνά πιο χρήσιμη για έναν χειριστή του δικτύου, σε σύγκριση με την παροχή εκτιμήσεων χωρίς πληροφορίες για το σφάλμα εκτίμησης. Η διαθεσιμότητα εκτιμήσεων υδραυλικής κατάστασης επιτρέπει επιπλέον την εκτίμηση της ποιότητας του νερού, λόγω της διασυνδεδεμένης φύσης της υδραυλικής δυναμικής και της δυναμικής ποιότητας του νερού. Παρέχεται μια μέθοδος εκτίμησης κατάστασης της ποιότητας του νερού η οποία υπολογίζει τα όρια της συγκέντρωσης χλωρίου και είναι σε θέση να ενσωματώσει αλγόριθμους για εκτίμηση των ποιοτικών παραμέτρων σε πραγματικό χρόνο, λαμβάνοντας υπόψη την ύπαρξη άγνωστων παραμέτρων ποιότητας νερού.

Ο κύριος στόχος των προτεινόμενων πρακτικών προσεγγίσεων εκτίμησης κατάστασης είναι να καταστεί δυνατή η ανάπτυξη μεθοδολογιών για τη διάγνωση σφαλμάτων στα ΣΔΝ. Ο υπολογισμός αξιόπιστων ορίων για τις εκτιμήσεις κατάστασης σε πραγματικό χρόνο αποτελεί βασική συνιστώσα πολλών μεθοδολογιών που σχετίζονται με τη διάγνωση σφαλμάτων χρησιμοποιώντας μοντέλα. Συγκεκριμένα, στοχεύουμε δύο τύπους κρίσιμων σφαλμάτων στα ΣΔΝ: τις διαρροές νερού και την μόλυνση του νερού. Προτείνονται δύο μεθοδολογίες για την ανίχνευση και τον εντοπισμό διαρροών με βάση την εκτίμηση εύρους τιμών για την υδραυλική κατάσταση του συστήματος και αξιολογούνται χρησιμοποιώντας ένα δείκτη αναφοράς διαρροής ανοικτής πρόσβασης και καθορισμένα δίκτυα αναφοράς. Η ανίχνευση μολύνσεων στο νερό επιτυγχάνεται συνδυάζοντας τις προτεινόμενες προσεγγίσεις εκτιμήσεως κατάστασης των υδραυλικών και ποιότητας νερού με μια λογική ανίχνευσης που χρησιμοποιεί κατώφλια πολλαπλών επιπέδων. Επιπλέον, διερευνάμε την έννοια της ενεργής ανίχνευσης βλαβών με την ανάπτυξη μιας νέας μεθοδολογίας για την επέκταση της περιοχής που παρακολουθείται από αισθητήρες ποιότητας νερού, η οποία αντιμετωπίζει το πρόβλημα της έλλειψης μετρήσεων.

Acknowledgments

This Ph.D. dissertation was performed under the supervision of Dr. Marios Polycarpou, Professor at the Department of Electrical and Computer Engineering of the University of Cyprus. I would like to express my deepest gratitude to my supervisor for giving me the opportunity to join the KIOS family and be in the middle of exciting innovation. His professional and wise guidance during my doctoral studies was essential in achieving the goals of this thesis. Most of all, I'd like to thank him for believing in me.

My special thanks goes to my mentor and friend Assist. Prof. Demetrios Eliades. Our collaboration throughout the years has made me a better researcher and a better person. His vision and forward thinking are what made this thesis possible.

I would like to express my sincere thanks to Assist. Prof. Stelios Timotheou and Prof. Avi Ostfeld for their guidance and scientific insights during our collaboration. Moreover, I would like to thank all my committee members and especially Prof. Christos Panayiotou and Prof. Symeon Christodoulou for their time to review and handle this Ph.D. dissertation.

Being a researcher at KIOS has granted me the opportunity to work in a motivating and pleasant environment while pursuing my Ph.D. Most importantly, it has given me life-long friends. Special thanks to Alexis Kyriacou, Panayiotis Papadopoulos, Marios Kyriakou and Lenos Hadjidemetriou for our helpful discussions and their support.

I dedicate this dissertation to my parents Chrystalla and George. The sacrifices they have made for me are the reason I have ever achieved anything in my life. They have taught me more than they know; how to love unconditionally, how to be an authentic person and the nobility of hard work. Thank you for always being there for me.

Last but not least, I want to express my gratitude to my lovely wife Maria for standing by me during this difficult journey, and to my daughter Iria for coming into my life. You two are the light which can guide me out of even the darkest places.

Publications

Journal articles (published)

1. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Real-time hydraulic interval state estimation for water transport networks: a case study,” *Drinking Water Engineering and Science*, vol. 11, no. 1, pp. 19–24, 2018.
2. S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, “Iterative hydraulic interval state estimation for water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 145, no. 1, p. 04018087, 2019.
3. S. G. Vrachimis, R. Lifshitz, D. G. Eliades, M. M. Polycarpou, and A. Ostfeld, “Active contamination detection in water-distribution systems,” *Journal of Water Resources Planning and Management*, vol. 146, no. 4, p. 04020014, 2020.

Journal articles (under review)

1. S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, “Leakage detection and localization in water distribution systems using interval model invalidation,” 2020 (submitted for review).
2. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Calculating chlorine concentration bounds in water distribution networks using real-time parameter estimation,” 2020 (submitted for review).
3. B. S. Rego, S. G. Vrachimis, M. M. Polycarpou, G. V. Raffo, and D. M. Raimondo, “State estimation and leakage detection of water distribution networks using constrained zonotopes,” 2020 (submitted for review).

Refereed conference proceedings

1. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Enhanced adaptive control of water quality in water distribution networks by incorporating abrupt hydraulic changes,” in *Proc. of Water Distribution System Analysis WDSA2014*, vol. 89, Bari, Italy, 2014, pp. 239–246.

2. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “The backtracking uncertainty bounding algorithm for chlorine sensor fault detection,” in *Proc. of Computing and Control for the Water Industry CCWI2015*, vol. 119, no. 1, Leicester, UK, 2015, pp. 613–622.
3. D. G. Eliades, D. Stavrou, S. G. Vrachimis, C. G. Panayiotou, and M. M. Polycarpou, “Contamination event detection using multi-level thresholds,” in *Proc. of Computing and Control for the Water Industry CCWI2015*, vol. 119, no. 1, Leicester, UK, 2015, pp. 1429–1438.
4. D. G. Eliades, M. Kyriakou, S. G. Vrachimis, and M. M. Polycarpou, “EPANET-MATLAB Toolkit : An Open-Source Software for Interfacing EPANET with MATLAB,” in *Proc. of Computing and Control for the Water Industry CCWI 2016*, Amsterdam, The Netherlands, 2016.
5. S. G. Vrachimis, M. S. Kyriakou, D. G. Eliades, and M. M. Polycarpou, “LeakDB : A benchmark dataset for leakage diagnosis in water distribution networks,” in *WDSA / CCWI Joint Conference Proceedings*, vol. 1, Kingston, Canada, 2018.
6. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Leak detection in water distribution systems using hydraulic interval state estimation,” in *Proc. of 2018 IEEE Conference on Control Technology and Applications (CCTA)*, Copenhagen, Denmark, 2018, pp. 565–570.
7. S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, “Interval state estimation of hydraulics in water distribution networks,” in *Proc. of European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 2641–2646.
8. R. Lifshitz, A. Ostfeld, S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Improving contamination detectability in water distribution systems using active fault detection,” in *WDSA / CCWI Joint Conference Proceedings*, vol. 1, Kingston, Canada, 2018.
9. S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Calculation of chlorine concentration bounding estimates in water distribution networks using real-time learning,” in *Proc. of Computing and Control for the Water Industry CCWI 2019*, Exeter, UK, 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In modern societies it is expected to open the valve of a faucet in our homes or elsewhere and clean water will run out. It is easy to forget that the quest for water was one of the defining struggles of human history. Civilizations that actively harnessed and sustained water, thrived, while the ones that failed, fell. An example is the Mayan civilization which, due to the tropical climate of the area and the naturally formed "cenotes", never developed infrastructure for water harnessing, and as a result it was decimated by a 100-year drought [16]. Today we are starting to see the effects of taking water for granted as we live in an era of rapidly growing human population. Cape Town was estimated to experience "Zero Day" in 2018, when taps in the city would run dry and the four million residents of the city would be required to collect daily water rations. This estimated date has since been postponed, when people in the city grew more conscious of the importance of water and started consuming less.

The growing public water conservation consciousness may increase the pressure to governments to invest more financial resources on the maintenance and efficient operation of water distribution systems. This aging critical infrastructure exhibits an increase of hydraulic failures such as leakages, pipe bursts, and malfunctioning valves and pumps. Mexico city, which faces a water crisis, loses 42% of their drinking water due to leaky pipes. In other areas like the island of Cyprus, where water shortages is a regular phenomenon, this number is close to 25%. It is estimated that every day more than 45 million m^3 of treated water is lost due to leakages in developing countries, which could have served 200 million consumers, and in addition, almost 30 million m^3 is consumed but not billed [17]. Moreover, hydraulic faults such as leakages may contribute to water quality deterioration, due to contaminant infiltration in the system.

Water sources are prone to pollution by the accidental injection, infiltration or illegal damping of sewage and agricultural waste into them. Sewage pollution is the ideal breeding-ground for the development of bacteria and viruses with staggering effects on health. According to a UN report, in 2010 microbial waterborne illnesses killed more people per year than war [200]. Authorities regularly monitor fresh water sources to ensure water quality standards are met and this valuable resource is safe for consumption according to the guidelines by the World Health Organization (WHO) [19]. However, after the water is pumped into the distribution network, monitoring of water quality is limited. Contamination events in drinking water distribution networks can be caused either by natural events, accidents or as a result of malicious attacks at any location in the network. For instance, in the town of Nokia, Finland, in November 2007, due to an accident, sewage water was injected into the town's drinking water distribution, affecting thousands of people and forcing the authorities to impose a complete ban on all water usage [199]. Industrial chemicals can also be a source of pollution. An example of such incident took place in West Virginia (USA) in 2014, where the water distribution network was accidentally polluted with crude MCHM (an industrial chemical), leaving almost 300,000 consumers affected [20].

It is obvious that we should protect this precious resource by investing in better management and modernization of our water distribution infrastructures. Modern technology is offering the opportunity to manage large critical infrastructures like never before. Sensors of various types are being installed in water systems as these devices become cheaper and more reliable. Sensor measurements are gathered by Supervisory Control and Data Acquisition (SCADA) systems making them instantly available to network operators, while also enabling remote control of system actuators. Moreover, integrated platforms are being developed that can relate sensor measurements with Geographic Information Systems (GIS) to provide a more complete view of the system. The advancements in modern communication technologies and the Internet of Things (IoT) have made economically feasible to receive large volumes of measurements and other available heterogeneous data with the aim to extract from them useful knowledge.

These developments have cultivated the vision of Smart Water Networks (SWN), which can be summarized by the following definition: Smart Water Networks refers to the use of sensing and communication technologies, along with intelligent algorithms for modelling, simulation, control, optimization and big-data analytics, for the purpose of enhancing efficiency and improving security, reliability, resilience, quality and robustness of drinking water distribution systems, as well as to minimize the impact of unforeseen events.

The realization of SWN relies on heterogeneous communication network architectures

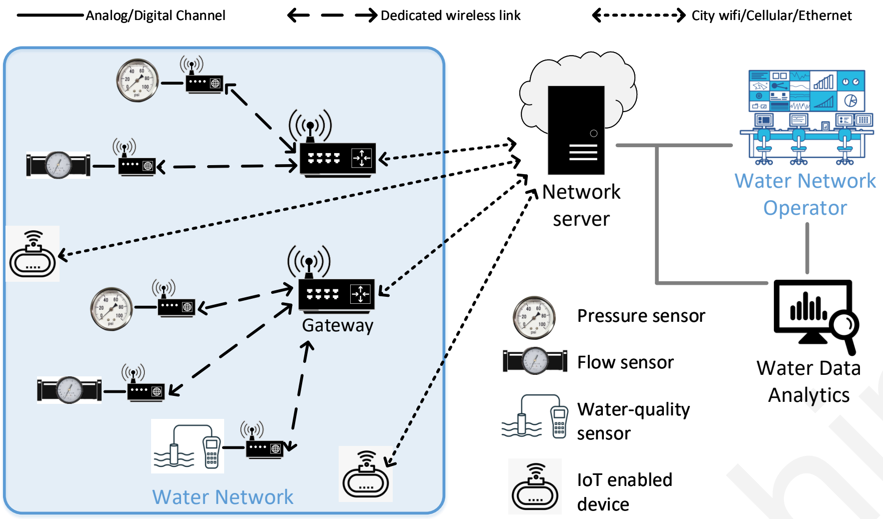


Figure 1.1. Communication network architecture for realizing smart water networks.

that interconnect the physical space (e.g., water distribution network) with the operational center for supporting all respective decision making processes. A typical architecture for such a network backbone is presented in Fig. 1.1, highlighting different types of connectivity and roles in the network; sensors deployed within the field of interest, network server for collecting and processing water-relevant data towards dedicated services (e.g., water network administration and water data analysis), and gateways which act as the communication bridges between the sensors and the network server. Sensors can exchange communication for network management and data relaying through underground links, while the data collected on the gateways can in turn be collected through underground-to-aboveground communication links. The architecture additionally considers command and control information originated from the network server/gateways to the sensors or network actuators.

At the heart of the required technologies for SWN is the real-time monitoring of water systems, which is the estimation in real-time of the hydraulic-state (e.g water flows) and the water-quality state (e.g., concentration of disinfectant, water age, the presence of a contaminant) in the network. State estimation can be used to better control and optimize the operation of the system. For example, pressure estimates can be used to minimize the overall pressure in the system by calculating optimal set-points for Pressure Reduction Valves (PRVs), while ensuring that pressure is sufficient to satisfy consumer demands. This feature significantly reduces the energy consumption of the network, while it also reduces pressure-dependent leakages. The most critical benefit of state estimation is that it enables the development of methodologies for early detection and prevention of unwanted events such as equipment failure, pipe breaks, and water contamination.

There are certain difficulties that need to be overcome in order to achieve the goal of effectively monitoring water infrastructures. For example, the appropriate policies should be enforced as authorities are generally reluctant to invest in the improvement of an infrastructure that, at least sub-optimally, works; until it doesn't [20, 199]. Additionally, there are significant practical difficulties in monitoring such a large-scale infrastructure, such as the prohibitive cost of installing and maintaining sensing devices in all the required locations and the partial knowledge of the system physical properties. Moreover, state-estimation as well as event detection, require the development of intelligent data processing algorithms specifically designed for the use in water systems.

In order to better understand the difficult task of real-time state estimation in WDS, a general description of these systems is given. Physically, a WDS is a collection of pipes which brings water to consumers from water sources, such as dams and reservoirs. The topology of a WDS is described by a graph of which the links represent pipes, pipe valves, and water pumps, with the last two being the main hydraulic actuators in a water network. The graph nodes represent junctions of pipes, consumer water demand locations, reservoirs, tanks and chemical dosing pumps with the latter being the water-quality input to the network. The dynamics of the system are typically divided into the hydraulic dynamics and water-quality dynamics.

1.1 Hydraulics description and challenges

Hydraulic dynamics describe the change in hydraulic-states, namely the hydraulic-head at nodes and the water flows in links. The hydraulic head is a measure of energy above a geodesic datum, which considers the elevation and the pressure at a node location, or the elevation and water level at a tank location. The inputs which drive the hydraulics are the actions from hydraulic-actuators such as pumps and valves. The biggest effect on hydraulics comes from consumer water demands which are modeled as an uncontrollable and typically unknown input.

The network is usually divided into subnetworks to enable better management. Large pipes which bring water from sources (like water reservoirs, dams and desalination plants) to cities and villages, are called the transport network. These contain large pumping stations which give enough pressure to send the water over long distances to the local water utilities. The utilities then further partition their network into consumer areas, called District Metered Areas (DMAs), which typically have only one inlet location at which the flow and pressure is measured. The greatest advantage of this partitioning is the ability to regulate the pressure at DMA inlets using Pressure Reduction Valves (PRVs), while there is also the added benefit of improving the visibility of leakages in each area. The layout of a typical WDS is illustrated in Fig. 1.2.

The physical structure of the networks is typically known to water utilities. Using GIS, the layout and properties of pipes are associated with the geography of the service area. The location of consumers in DMAs are the most uncertain component of this representation. To pinpoint these locations, utilities manually record the geographical coordinates at meter locations and include them into the GIS of the network. The recent EU 'Open Data Directive' [21] which urges governmental agencies to adopt Open Data policies, benefits water utilities, as well as practitioners, in the task of developing more accurate models of their systems. For example, by combining open data from buildings, water utilities can deduce consumer demand locations which have been omitted from the GIS model.

Modeling the physical structure of a WDS in GIS, enables the creation of hydraulic models, which is a mathematical representation of hydraulic dynamics in the distribution network governed by the laws of mass and energy conservation. Hydraulic models are formulated for each system based on its structural properties, such as the network graph, pipe lengths, diameters etc., and assuming known hydraulic inputs, i.e., pump status, valve settings and water demands. An accurate hydraulic model can be used to determine unknown states of the network, such as the water flows in pipes. Some properties are difficult to be defined accurately, such as pipe roughness, a property which describes the resistance exhibited by pipe walls to water flow. Moreover, while some inputs (such as pump and valve settings) are typically known, others (such as consumer demands) are unknown.

Traditionally, the main sensing devices present in a WDS are flow meters at pumping stations, utility reservoirs and at DMA inlets, while pressure sensors may also exist at these locations. These devices give measurements of flow or pressure at regular time intervals which range from a few minutes to hours. At the DMA level, demand volume meters are installed at the majority of consumers which are manually logged by water utilities every few months and due to this low measurement resolution they cannot be directly used for real-time state estimation inside a DMA. However they are useful at deducing an average

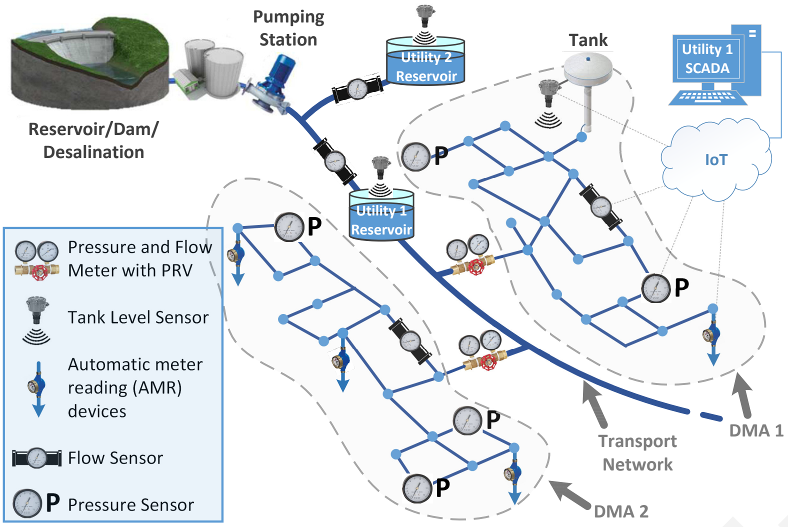


Figure 1.2. Layout of hydraulic sensors in a smart water network.

demand for each consumer. For residential areas, this average estimate (or base demand) can be improved by combining open data about population densities and distribution as well as building locations and size. The base demands at locations without available measurement data can be calculated by multiplying the average consumption per person by the estimated number of people per building.

An important component of the modernization of WDS involves the installation of sensors inside DMAs. For instance, water level sensors may be installed in tanks, flow sensors on pipes and pressure sensors at nodes, with the latter being more attractive to water utilities due to lower installation and maintenance costs [22]. These devices are typically connected to a telecommunications infrastructure to send data at a relatively high resolution (minutes) at the water utility SCADA station. Moreover, Automatic Meter Reading (AMR) devices may be installed which remotely monitor the consumption of individual consumers. AMRs are installed to monitor large numbers of consumers and simplify the billing process. They typically send low resolution measurements to minimize communication cost (cost per message through GSM communications infrastructure) and maintenance cost (replacement of batteries). These devices send water volume readings, instead of flow readings required by hydraulic state-estimation methodologies and, as a result, considerable measurement uncertainty is added during the unit conversion process. The layout of sensors in a modern WDS can be seen in Fig. 1.2

Transport networks are parts of WDS which are ‘well’ monitored, i.e., the inlets of this sub-network are measured (typically receiving water from sources or reservoirs), and so are the outlets (utility reservoirs or DMA entrances). These measurements enable the complete hydraulic-state estimation in a transport network in combination with a hydraulic model. Generally, to achieve the complete hydraulic-state estimation of a water network, at least as many sensors as the number of inlets and outlets of the network should exist. These sensors do not necessarily need to measure flow at inlet and outlets, but also flow and pressure at other locations. They do however need to be in a configuration which allows the solution of hydraulic equations without these becoming under-determined. This concept is referred to as *topological observability* [23].

On the other hand, achieving topological observability in DMAs is challenging because DMAs have hundreds or thousands of water outlets, i.e., demand locations, which would mean the need for installing thousands of sensors in every DMA. Even with the installation of high resolution AMRs at all consumers there would be significant errors in state estimation due to unregistered water consumption and background leakages. The distribution networks of DMAs, however, are the parts of water systems which experience the most pipe failures and the most water loss [17]. Moreover, water contamination in these areas will have an immediate impact on consumers. It is evident that the monitoring of these areas is essential and new ways of doing so is a subject of research.

A tool used by researchers to reduce the required number of sensors in DMAs is creating models with a *reduced state-space*. One way to do this is by clustering many demand points in a single node, and thus formulating a reduced model [24]. In Fig. 1.3 a clustering procedure is illustrated, in which a reduced GIS model is created by grouping many demand locations (buildings) in a single node using geographical polytopes. Clustering is performed with the aim of preserving the hydraulic characteristics of the network, e.g., nodes should exist at pipe junctions where the diameter changes and at valve locations. Moreover, segments of pipes with consumers should be modeled by at least one demand node. This way of creating a reduced model retains the physical structure of the network, thus the model can be directly associated with the actual system. However, there is a limit on the degree of reduction before losing this direct association, and typically observability in the reduced model is not achieved with the available sensors. Additionally, some structural modeling uncertainty is introduced due to the mismatch and reduced number of demand locations.

Other approaches for model reduction use a procedure called *skeletonization* which changes the structure of the network by removing pipes as well as nodes and changing the remaining pipe properties. This results in a skeletonized version of the network which is only implicitly related to the real system, however its state-space is reduced significantly. The modeling un-

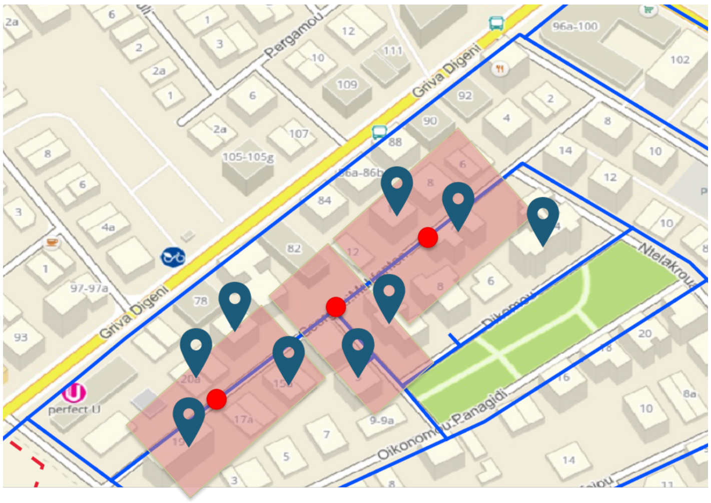


Figure 1.3. Grouping of consumers into nodes in a GIS using building open data.

certainty introduced is now considerably larger compared to methodologies that only remove nodes, and it is not straightforward to relate the model with the actual physical system.

In summary, there is an opportunity for hydraulic state-estimation due to the advancements in sensors and communications technology and the developments of integrated platforms and GIS models using also open data. The challenges are how to use reduced models which contain significant uncertainty in combination with limited sensor measurements which do not guarantee observability to achieve hydraulic state-estimation in all the parts of the network, including DMAs.

1.2 Water-quality description and challenges

Water quality in WDS refers to the concentration of different substances of interest in the water. These substances may be contaminants (metals, chemicals, pathogens) which are harmful to humans, such as lead originating from aging and deteriorating pipes or arsenic originating from groundwater sources, or disinfectants inserted intentionally by water utilities with the aim to kill pathogens. The most common disinfectant, chlorine, is highly effective in destroying water's microbiological organisms (see coronaviruses), and a minimum chlorine residual is required to be present throughout the network for disinfection purposes. Due to the fact that chlorine produces certain disinfection by-products, such as trihalomethanes (THMs) [25], which are linked to some types of cancer when consumed over a long period of time at high concentrations, the chlorine concentration should be limited within certain bounds, as described by EU directives and national legislation. According to the WHO, a free chlorine residual concentration must exist in WDS, with minimum target concentration $0.2\ mg/L$

at the point of delivery and 0.5 mg/L for high-risk circumstances [19]. Moreover, if over-chlorination occurs during the disinfection process, it could trigger other chemical reactions, causing pipe corrosion and the release of iron, copper and lead in the drinking water [26].

Recent EU directives [27] emphasize the importance of water-quality monitoring for reducing the risks to health from drinking water, with chlorine and its by-products are among the quality parameters recommended for monitoring. Real-time estimation of chlorine-concentration state in a WDS can be used for better regulation of this disinfectant and reduce its by-products. Additionally, due to the fact that certain contaminants will affect chlorine residuals in a specific way (e.g., a bacterial toxin may decrease the concentration of free chlorine due to its reaction dynamics) [28], chlorine residuals can be used as indicators of contamination events [29]. The aforementioned indicate that chlorine concentration monitoring can be a useful tool for determining water quality.

Monitoring of chlorine concentration in a complex infrastructure like WDS requires the use of models which consider the transport of chlorine from injection points to consumers, and also its reactions with other substances in the water. The input to a chlorine concentration model are the chlorine injection points placed typically by water utilities in reservoirs and tanks. Chlorine is then transported to different parts of the network through pipes, following the water flow dynamics. The nature of hydraulic and water-quality dynamics are inherently interconnected and specifically the hydraulics affect water-quality by the varying water flow in pipes. This concept is illustrated in Fig. 1.4, where it is easier to see that water-quality states depend not only on the injected chlorine, but also on consumer water demands and hydraulic actions. The time-varying nature of the, typically unknown, consumer demands results in water-quality models that exhibit time-varying behavior. Chlorine reactions are affected by the presence of bacteria or chemicals in the water, but also on materials accumulated on pipe walls. It is typically assumed that the reaction rates of chlorine in drinking water remain relatively constant, however, this is not the case in real systems. Chlorine reaction rates may vary, affected by water temperature, pH , organic content, etc. [30], thus contributing to the time-variability of the model.

The common practice for water-quality monitoring is to periodically manually take samples from certain locations in the distribution network and analyze them for various substances. This practice is not effective in the case of contamination events since many customers will be affected before such event is confirmed. Due to the lack of real-time measurements, the control of disinfectant residuals is performed in an open-loop approach. This means that the amount of disinfectant which is inserted into the system is calculated based on

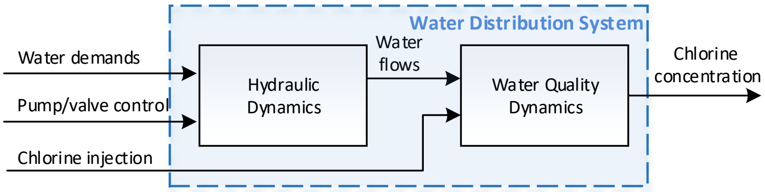


Figure 1.4. The interconnected nature of hydraulic and water quality dynamics in a Water Distribution System.

expert knowledge and off-line data, an approach which often leads to over- or under-dosing.

An increasing trend is to employ water-quality sensors which can monitor on-line multiple generic water-quality parameters such as chlorine concentration, turbidity, conductivity and pH, many of which are affected in the case a contaminant enters the water. The location of the sensors may be selected after solving the optimal sensor placement problem [31], which typically tries to minimize the impact of a potential contamination by maximizing water-quality monitoring coverage spatially and temporally. These sensors typically do not detect the presence of contaminants directly, rather than their effect on the measured water-quality parameters, such as chlorine concentration. The task of detecting the effect, and in extent the existence, of contaminants on water-quality states such as chlorine concentration is challenging as it requires the estimation of normal chlorine concentration in the absence of contaminants; a difficult task to achieve given the small number of sensors and the uncertain, time-varying water-quality model. An added benefit of the real-time estimation of chlorine concentration is that it enables booster chlorination [32]. Booster chlorination stations are installed in a spatially distributed manner in the system and collectively insert less chlorine compared to a single input location at the inlets, thus also significantly reducing the formation of chlorine by-products. Without chlorine concentration estimation however, chlorine control methodologies may perform poorly leading to under- or over-chlorination. The layout of water-quality sensors and chlorine injection stations in a modern WDS can be seen in Fig. 1.5

In summary, recent legislation by the European Union has imposed standards on water quality that water utilities should follow, in order to guarantee the safety of drinking water. The installation of water-quality as well as hydraulic sensors in WDS offers the opportunity to monitor and regulate water-quality in real-time, significantly reducing the risks to human health. The challenge is to achieve real-time water-quality monitoring using a limited number of sensors and highly uncertain time-varying water-quality models.

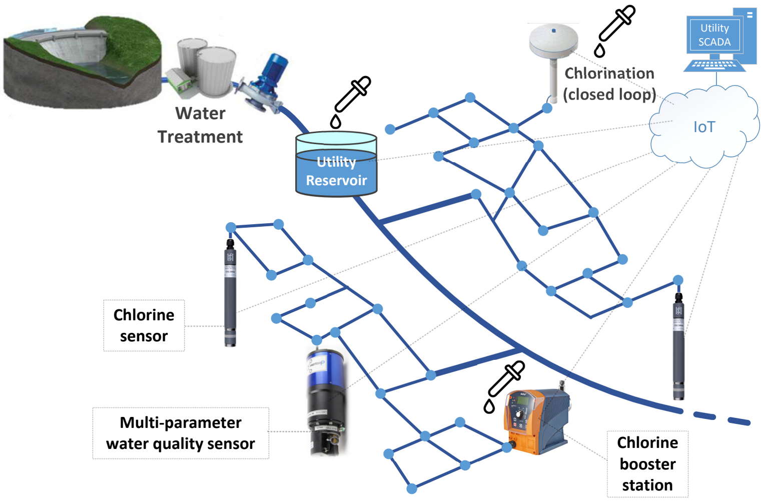


Figure 1.5. Layout of water-quality sensors in a smart water network.

1.3 Thesis Objectives and Contributions

The first general objective of this thesis is to develop methodologies for state estimation in WDS that take into account the main practical difficulties of this attempt and are designed to give a reliable state estimate, which is also useful to water utilities and practitioners. To ensure reliability, modeling uncertainty and measurement errors should be considered. To provide a useful estimation, novel approaches should be developed which use *a priori* available information, due to the limited measurements available. The aim of these practical state estimation approaches are to enable the development of methodologies for fault diagnosis in water distribution systems, which is the second general objective of this thesis. Specifically, two critical faults in these systems are targeted: water leakages and water contamination. Considering the above challenges in WDS and the general objectives, the key contributions of this thesis are summarized as follows:

- The development of a novel interval-state estimation algorithm that calculates tight hydraulic-state bounding estimates considering known bounds on parameter and measurement uncertainties. The algorithm uses the nonlinear form of the hydraulic equations, considering also pressure-dependent demands and background leakages, in order to ensure that the bounding estimates include the true system state, given that uncertainties have been accurately represented. The algorithm is combined with an interval observer which uses a state-space modeling of hydraulics to calculate bounds on the dynamic states.
- The development of leakage diagnosis methodologies based on the developed interval-

model and interval-state estimation algorithm. First, a leakage detection methodology is developed which uses the interval-state estimates directly as thresholds to formulate the leakage detection problem. Then, a novel methodology for leakage detection through interval-model invalidation is developed that demonstrates increased detection rates compared to the previous methodology. Finally, an optimization-based methodology for leakage localization using the proposed interval-model is developed which yields a priority list of possible leakage nodes.

- The development of a model-based algorithm to calculate chlorine concentration bounds at nodes of WDS, using the backtracking approach. The methodology is combined with a real-time parameter estimation algorithm for bulk reaction coefficients of water originating from different sources and it is tested using real data from a transport network.
- The development a model-based contamination detection methodology for WDS using chlorine concentration thresholds. A Monte-Carlo simulation based approach is used to create multi-level thresholds, for enhancing detection and reducing detection delay while minimizing false positive alarms. The methodology is improved by using the analytically derived chlorine concentration bounds to create the detection thresholds.
- The development of a novel approach based on the concept of *active fault detection*, which alters the area monitored by water-quality sensors by manipulating the system actuators in order to increase contamination detectability, given the scarcity of water-quality sensors.
- The development of software tools for WDS analysis. Specifically, in the context of this work, the EPANET-MATLAB toolkit was developed which interfaces the libraries of the established water system simulation software EPANET with the MATLAB programming language. Moreover, an open access leakage diagnosis benchmark is developed for the assessment of leakage diagnosis algorithms.

1.4 Thesis Outline

In the following, we present the outline of each chapter:

Chapter 2: Literature review

In this chapter we outline the state-of-the-art in four distinct areas of interest:

1. *Hydraulic-state estimation*, exploring techniques which deal with the specific uncertainties in WDS;

2. *Leakage diagnosis*, giving particular emphasis on model-based methodologies;
3. *Water-quality state estimation*, exploring water-quality modeling and methodologies which try to estimate chlorine concentration;
4. *Contamination detection*, outlining current passive contamination detection techniques.

Chapter 3: Hydraulics modeling considering uncertainty

This chapter formulates the hydraulic model a WDS considering parameter and measurement uncertainty. First the hydraulic equations derived from the laws of conservation of mass and energy in these system are formulated. The available measurements in these systems are then analyzed and selected to obtain observability. Uncertainty is modeled using bounds on uncertain parameters and measurements, due to the need of using artificial demand estimates called, pseudo-measurements, which are highly uncertain and without a statistical characterization of the mismatch to the real value. Modeling uncertainty is also considered in the form of uncertain pipe parameters. Moreover, a state-space formulation of hydraulics is formulated to consider the dynamics when tanks exist in the system.

Chapter 4: Hydraulic interval state estimation

In this chapter we propose a novel interval hydraulic state-estimation approach for WDS that considers the combined effect of bounded measurement and modeling uncertainties. The proposed *Iterative Hydraulic Interval State Estimation* (IHISE) methodology calculates bounds on state estimates using the nonlinear form of the network algebraic equations and considering pressure-dependent demands. The nonlinear modeling guarantees that when accurate uncertainty bounds are provided, the bounds on state estimates will include the true system state. This is achieved using *bounding linearization*, a technique which restricts the nonlinearities within a convex set, thus converting the hydraulic equations in a form where the minimum and maximum of each state can be found using linear programming. An iterative procedure minimizes the distance between upper and lower state bounds, by reducing the feasible set defined by bounding linearization at each step and converging to the tightest possible bounds. Moreover, the state-space hydraulic model is used to design an interval observer for calculating bounds on the dynamic states.

Chapter 5: Leakage detection using interval-state estimation

In this chapter we describe a methodology of using hydraulic interval-state estimation for the detection of leakages in WDS. The methodology is applied on a case study of a modified transport network in Cyprus, using real data, with the aim to determine the existence of unaccounted-for water. The results show that the existence of an artificial leakage into the data can be detected, due to state bound violation. Furthermore, the methodology is evaluated using the Leakage Diagnosis Benchmark (LeakDB) dataset developed in the context of this work.

Chapter 6: Leakage diagnosis using interval-model invalidation

In this chapter we propose the utilization of *a priori* available information about the water distribution system to formulate a hydraulic model in its non-linear form, in which uncertainties are modeled by intervals defined by a lower and upper bound. A novel optimization-based methodology then utilizes pressure and flow measurements to perform leakage detection through model-invalidation. A modification of the optimization algorithm is activated in the case of a detection to refine possible leak locations and retain only the ones that can be explained by the interval model and available measurements from multiple time-steps. The proposed methodology is demonstrated on a benchmark network and evaluated using a leakage diagnosis benchmark dataset.

Chapter 7: Bounded water quality state estimation

In this chapter we propose a methodology which calculates chlorine concentration bounds at node locations of water distribution networks and is suitable for sensor-fault and contamination detection purposes. The proposed *Backtracking Uncertainty Bounding Algorithm* (BUBA) is able to calculate these bounds, given bounds on hydraulic-states produced by hydraulic interval-state estimation and bounds on water-quality model parameters. The validity of the calculated bounds is demonstrated using a large number of Monte-Carlo simulations on benchmark networks. Moreover, the proposed methodology can be used in conjunction with real-time learning algorithms, as it allows for time-varying model parameters. A parameter estimation algorithm is designed and implemented to learn bulk reaction coefficients of water originating from different sources. The BUBA then uses the time-varying parameters to improve the chlorine concentration bounds, as demonstrated using a case study of a real water transport network.

Chapter 8: Contamination detection using chlorine concentration thresholds

In this chapter, chlorine bounds are calculated using Monte-Carlo Simulations (MCS) and used as contamination detection thresholds. Moreover, a methodology which uses multi-level thresholds is proposed, which facilitates the computation of a confidence metric regarding the occurrence (or not) of a contamination event. We show that by combining the analytical methodologies for hydraulic interval-state estimation and bounded water-quality state estimation of Chapters 4 and 7 respectively, we can achieve the creation of the same multi-level thresholds, eliminating the need for MCS. Moreover, the use of current measurements by these methodologies improve the generated bounds.

Chapter 9: Active contamination detection

In this chapter we address the problem of water-quality sensor scarcity by proposing a novel methodology for altering the area monitored by water quality sensors in Water Distribution Systems (WDS) when there is suspicion of a contamination event. The proposed Active Contamination Detection (ACD) scheme manipulates WDS actuators, i.e., by closing and opening valves or by changing the set-points at pressure controlled locations, to drive flows from specific parts of the network in predetermined paths, and enable the sensors to monitor the quality of water from previously unobserved locations. As a consequence, the monitoring coverage of the sensors is increased and some contamination events occurring within those areas can be detected. The objective is to minimize the contamination impact by detecting the contaminant as soon as possible, while also maintaining the hydraulic requirements of the system. Moreover, the methodology facilitates the isolation of the contamination propagation path and its possible source. We demonstrate the ACD scheme on two networks, analyze the results and open the discussion for further work in this area.

Chapter 10: Conclusions

In this chapter we present some concluding remarks, as well as directions for future research.

Chapter 2

Literature review

2.1 Hydraulic state estimation

Monitoring of hydraulics in a WDS requires the existence of hydraulic-state estimation algorithms which are able to infer the complete system state, namely the water flows in pipes, pressures at nodes, and water level of tanks, using the available measurement set and network model. A complete view of the network state supports the decision-making process and enables the efficient operation of these systems, improving customer service. Moreover, it enables the timely response to undesirable events, such as pipe breaks or failure in equipments such as pumps, thus minimizing their impact. Examples of the use of state estimation in real systems include the use for online burst detection [33], online modelling [34] and control of WDS [35].

Standard state estimation techniques require a measurement set that makes the system observable, i.e., the sensor number and locale ensure that the system state can be calculated [36, 37, 38]. Additionally, the statistical characterization of sensor measurement error is needed to give more weight to measurements originating from more accurate sensors. Then, using a mathematical model of the network, a state estimation algorithm can infer the system state. Many approaches have been proposed to solve the state estimation problem for water systems, such as Kalman Filtering and Weighted Least Squares (WLS), with the latter being the most widely used and varied. The above methods produce a point in state-space and are referred to as *point state estimation* [39, 40, 41].

State estimation in WDS is a challenging task, mainly due to the scarcity of measurements which are not enough to achieve system observability. Some parts of the WDS may be widely monitored, such as transport networks, however even a single sensor failure could make these parts unobservable [1]. The large area covered by WDS and the large number of system states is one of the main reasons that a large number of sensors must be installed to guarantee observability. A common practice to reduce the complexity, is to use reduced models by treating a group of consumers as a single demand point. It is then possible to use *pseudo-measurements*, which are demand estimates determined from population densities and historical data, to complement the missing measurements [42]. Recent advances in water demands research have also made possible the higher resolution modeling of water demands, thus reducing the need for skeletonization of networks and increasing the accuracy of state estimation [43].

The use of pseudo-measurements may introduce new problems to the state estimation process, as they are highly uncertain and the resulting estimates may deviate significantly from the real system state. This in turn could affect other algorithms which rely on state-estimation, such as feedback control or fault-diagnosis. Efforts have been made to characterize the uncertainty of pseudo-measurements [44], but it is improbable that a statistical characterization will be available. Thus, standard state estimation techniques such as WLS may not be capable of producing a reliable measure of the estimation error. Consequently, researchers have tried to combine online estimation of demands with state estimation, in order for the latter to be more accurate [24].

Demand calibration is the use of available historical sensor measurements and the system model to approximate using optimization based methodologies the demand magnitude and behavior at nodes [45]. It is considered as the inverse problem to hydraulic-state estimation [46]. Similarly to hydraulic-state estimation in which the scarcity of measurements does not guarantee observability [47], the inverse problem of demand calibration is ill-posed and it has attracted a lot of attention by researchers [48, 49]. Typically the under-determined problem of demand calibration is dealt by grouping several demand node parameters to make the problem over-determined. This produces an accurate estimate for the group however the individual node estimates may have significant errors [49]. The accurate estimation of group demands was exploited in [50] where the authors propose a leak-detection and localization approach coupled with a calibration methodology that identifies geographically distributed parameters. Due to the high number of tuning parameters, heuristic optimization techniques are used which may not produce reliable results [51].

Another significant source of uncertainty which complicates WDS state estimation is modelling and parameter uncertainty. Recent works provide explicit expressions for the sensitivity of the state estimation problem to these uncertainties [52]. Typically, the network topology is assumed known, especially after the process of skeletonization which simplifies the network graph. However, even when the topology is known, pipe parameters such as length and diameter are rarely known accurately and estimates are used in place. This is especially true for pipe roughness coefficients, which along with pipe length and diameter, are used to calculate head-loss across pipes. This is why, even with an observable sensor configuration, model calibration is required *a priori* or during state estimation for the latter to produce feasible solutions [53]. Due to the large number of unknown parameters with respect to measurements, it is possible for the calibration process to calculate parameter values which satisfy the constraints imposed by measurements, but deviate from the true parameter values.

Considering the many unknowns and uncertainties in WDS state-estimation, it is evident that accurate state-estimates are difficult to be generated without some kind of trade-off. A practical approach for state estimation in the presence of demand and modeling uncertainty, is *interval state estimation* [54, 55]. This approach models the uncertainties on input data as intervals, defined by lower and upper bounds. Then, considering this bounded uncertainty, interval state estimation provides lower and upper bounds on the state estimates, in contrast to point state estimation methods which only provide a single point. Providing a range of values for each state, is often more useful to an operator than providing point estimates which give no indication on their proximity to the true state value. Additionally, having reliable interval state estimation is essential in many methodologies related to event and fault detection such as leakage detection [56], water contamination detection [9] and sensor fault detection [8].

The use of bounds for the representation of measurement uncertainty and the subsequent calculation of state estimate bounds was introduced in [44]. This idea was further developed in [57] as the *set-bounded* state estimation problem. The process of calculating bounds for state estimates caused by measurement uncertainty is also referred to as Confidence Limit Analysis which can be solved using different approaches, including Neural Networks [58], the Error Maximization method [59], the Ellipsoid method and Linear Programming [54]. All these approaches assume a known network model which can be linearized in order to solve the non-linear equations that characterize WDS and provide state bounds based on measurement uncertainty. Few methodologies can guarantee the inclusion of the true state in the bounds based on given uncertainty, while the effect of modeling uncertainty is not considered. Another approach that could incorporate modeling uncertainty is the use of Monte-Carlo Simulations (MCS), where state bound estimates are obtained by randomly generating and evaluating a large number of model parameter sets or realizations [60]. This approach requires a sufficiently large number of simulations, and even then some cases may not be covered, leading to underestimation of the range of the true state bounds. Moreover, it is not trivial to use the available measurements to constrain the outcome of simulations, thus typically only the inputs of the system are used.

2.2 Leakage diagnosis

Leakage diagnosis in water distribution systems has attracted a great deal of attention from both practitioners and researchers over the past years [61], while it is also the topic of the latest *Battle of the Leakage Detection and Isolation Methods* (BattLeDIM) [62]. It can be separated into leakage detection methodologies which focus on identifying the existence of a leak in the network, and leakage localization which aims to provide an approximate location of leakages given the available measurements. A recent review paper [61] provides a classification of current leakage detection methodologies as A) Passive methods and B) Active methods.

Passive methods (also referred to by other researchers as equipment-based, hardware or external methods) require the deployment of special equipment at an area which is suspect of leakage. Such methods involve acoustic equipment such as listening rods, leak correlators, leak noise loggers and non-acoustic methods like gas injection, ground penetrating radar technology and infrared photography [63]. These methods have proven very effective in localizing pipe bursts, however they come with significant disadvantages such as requiring a long time to find a leak in a large search area, needing a large number of sensors, being invasive, labor-intensive and causing disruption of service [64, 63].

Active methods (also referred to by other researchers as internal or software methods) are methods which assume the existence of permanently installed sensors which continuously monitor the system for leakages. The latest developments in hydraulic sensor technology and on-line data acquisition systems have enabled water companies to deploy a larger number of more accurate pressure and flow devices with less cost. These data can be used to monitor the system in real-time and develop methodologies that use the data to detect and pre-localize leaks. Pre-localization is the process of defining an area in which the leak exists, instead of pin-pointing exactly its location. This area of research has seen a significant interest in recent years, as seen in recent review papers [63, 61, 65]. Depending on the way active methods process the acquired data, they are subdivided into three categories: Transient model-based, data driven and model-based methods.

Transient model-based methods [66, 67] analyze pressure transients which appear after the occurrence of a leak using pressure sensors with high sampling rate. The transient model is very sensitive to the configuration of the whole network, and is obtained using the risky and unwanted process of generating a “water hammer” in the system [68]. The aforementioned drawbacks, along with the cost of high frequency sensors, are the main reasons why these methodologies are only used on individual large pipelines [48].

Data-driven methods (also referred to as non-numerical model methods) do not require a model to perform detection. They do required large amounts of reliable training data and a drawback of these techniques is that they can’t be used when this data is not available [63]. An example of a data-driven approach is found in [69] where the authors first introduced artificial neural networks (ANNs) for burst detection and have continued to extend their work in the following years [70]. Another approach is found in [71] where the authors proposed an algorithm which analyzes the discrete inflow signal of a DMA by using an adaptive approximation methodology for updating the coefficients of a Fourier series and detects leakages by utilizing the Cumulative Sum (CUSUM) algorithm. To improve performance the authors in [72] used a mixed model-based and data driven approach, to improve performance. The study in [73] provides a review on data-driven approaches for burst detection and concludes that are promising in real-life burst detection, but reducing false alarms is still an important issue and a comprehensive performance evaluation, especially under different network configurations, might be necessary.

The term *model-based leakage diagnosis* is used to describe methodologies that utilize a model of the WDS (also referred to as numerical model) to estimate the steady-state hydraulic conditions in the network. This steady-state model neglects transients [67] and it can be obtained using known physical properties of the system. As a result, it is the most widely used model in water distribution systems analysis [74]. Typically, model-based methods assume the existence of a larger number of pressure sensors than flow sensors in WDS because they are cheaper and easier to install and maintain [75]. However, WDS are large-scale systems and the number of sensors is still limited compared to the system size. This is why methodologies for optimal placement of pressure sensors based on the available model are used to facilitate leakage diagnosis [76, 77, 78].

The operating principle behind model-based leakage detection, as first suggested by [79], is to find discrepancies of measurements to their estimates obtained by the network model, which would indicate the existence of a leakage. For localization, some approaches try to relate acquired measurements with the simulated output from many leakage simulation scenarios on different locations of the network [76, 80]. The geographical mapping of each model component can then be used to signal the probability of a zone to contain a leakage [22]. Researchers have also used pressure residual analysis, by creating a system pressure sensitivity matrix, to identify the location of leaks with the assumption of a single leakage occurring in the system [75, 78].

The aforementioned methodologies consider the availability of consumer demand estimates in the network model. The estimates, or pseudo-measurements, of these demands can be derived by utilizing historical billing data or data from smart meters. These data are essentially low resolution measurements which can only be used to compute an average base demand at nodes. The variation of base demands at different times of the day can be approximated by multiplying them with a pattern derived from the measured system inflow [50]. However, the resulting demand pseudo-measurements may contain significant errors with no information on their statistical characterization, which is why some researchers have argued that the most accurate depiction of demand pseudo-measurement error is in the form of error bounds [40]. In practice, consumer demands are the greatest source of uncertainty when using model-based methodologies for leakage diagnosis [80]. This is because demand variations can cause significant pressure variations, which may either hide the effect of a leakage (unless the leak size is large relative to the system inflow), or cause false positives [81].

The uncertainty in water demands makes *demand calibration* using sensor measurements an important component of model-based leakage diagnosis. Demand calibration refers to the use of available historical sensor measurements and the system model to approximate using optimization based methodologies the demand magnitude and behavior at various nodes of the network. It can be viewed as the inverse problem to hydraulic-state estimation [46]. Similarly to hydraulic-state estimation in which the scarcity of measurements does not guarantee observability [47], the inverse problem of demand calibration is ill-posed, and has attracted significant attention by researchers and practitioners [48, 49]. Typically, the underdetermined problem of demand calibration is dealt by clustering several demand node parameters to make the problem over-determined. This produces an accurate estimate of the grouped node demand, however the individual node estimates may have significant errors [49]. The accurate estimation of group demands was exploited in [50] where the authors propose a leakage-detection and localization approach coupled with a calibration methodology that identifies geographically distributed parameters. The geographical distribution of the calibrated parameters enables localization, as abrupt changes to these parameters are associated with a specific zone in the network. Leakages are in essence an unknown or unaccounted demand of the system and this is why some research approaches have considered leakage detection as part of calibration [82]. However, the joint demand and leakage estimation problem increases the number of tuning parameters and the heuristic optimization techniques used to solve this problem may not produce reliable results [83, 51].

In addition to demand uncertainty, model parameters such as pipe roughness coefficients, introduce additional uncertainty into the calibration and leakage diagnosis problems. Due to the relatively small number of measurements, demand and parameter calibration may be jointly performed [84]. An estimate of pipe roughness parameters can be calculated using the pipe specifications given by the water utility. Since these estimates contain errors more accurately depicted by error bounds, a recent approach considers the bounded model parameter and demand uncertainties to calculate bounds on state estimates [1].

In the fault diagnosis literature, modeling uncertainty is a key component in calculating the threshold on the residual; i.e., the difference between the measured and estimated states. When the threshold is violated by the residual, the occurrence of a fault is declared [85]. This concept is applied to leakage detection in [86] where a Linear Parameter Varying model is used to compute a set-valued state estimate. The method allows the creation of zonotopes, which are compared with the pressure residuals and are used as thresholds for detection. However, the uncertainty on water demands is not considered, while the use of a linearized model may introduce additional modeling errors. In [12] the authors investigated the leakage detection problem by comparing sensor measurements to thresholds derived from hydraulic interval-state estimation on a nonlinear model using demand pseudo-measurements. Although the particular algorithm is designed to achieve zero false-positives, in practice it has low leakage detection rate on a related benchmark data set [11]. The reason is that the algorithm obtains hydraulic state bounds without incorporating into the model available information such as state measurements and aggregate demand bounds.

2.3 Water-quality state estimation

The most recent EU directive on the quality of water intended for human consumption [27] emphasizes the importance of water quality monitoring for reducing the risks to health from drinking water, with chlorine by-products being among the quality parameters recommended for monitoring. Chlorine is commonly used by water utilities as a disinfectant in Water Distribution Networks (WDN). According to the WHO, a chlorine residual needs to be sustained throughout the drinking water network which should be sufficient to deactivate waterborne pathogens and, at the same time, small enough to reduce the formation of harmful chlorine by-products [25]. Additionally, due to the fact that certain contaminants will affect chlorine residuals in a specific way (e.g., a bacterial toxin may decrease the concentration of free chlorine due to its reaction dynamics) [28], chlorine residuals can be used as key indicators of contamination events [29].

Current practice of residual regulation is the insertion of pre-defined chlorine in water tanks in the network. This practice is inefficient in its ability to distribute the chlorine evenly throughout the water network, resulting in some areas having a low chlorine concentration in their supplied water and in other areas having a high concentration. Researchers have proposed the regulation of chlorine concentration through the insertion of chlorine injection stations in specific location in the network, known as chlorine booster stations. With a greater number of chlorine inputs in the network, the regulation of chlorine in any location of the network can be achieved [87].

A common practice for water quality monitoring is to manually take samples from certain locations in the distribution network. This practice is not effective in the case of a contamination event since many customers will be affected before such event is confirmed. The installation of on-line chlorine sensors can enhance monitoring capabilities of chlorine residuals and enable better control actions. For contamination event detection, water-quality sensors may be installed which can monitor multiple parameters (pH, free chlorine, total organic carbon) and detect changes in water quality. The placement of such sensors in WDN with the objective to minimize the risk of a contamination event has been a subject of research for years [88, 89]. The problem is posed as a multi-objective optimization, with one of the objectives being the minimization of the number of sensors due to their high capital and maintenance costs [90]. Single-parameter sensors (such as free chlorine concentration sensors) are cheaper than multi-parameter sensors, and using them not only for chlorine residual monitoring but also for contamination event detection, could significantly reduce the health risks from contaminated drinking water. However, in order to determine whether a sensor reading is abnormal (e.g., due to a contamination), it needs to be compared with an estimate of the expected concentration at the sensor location. The expected concentration can be calculated using water-quality models [91]. These are mathematical representations of the physical and chemical laws that describe chemical transport and reaction in WDN and are used to calculate the water-quality states (in this work, the chlorine concentration at nodes).

Chlorine concentration at a network location depends on the chlorine input, but also on consumer water demands and hydraulic actions which affect pipe water flows, as seen in Fig. 1.4. Due to the time-varying nature of the typically unknown consumer demands, water-quality also exhibits a time-varying behavior. Some approaches have exploited the apparent periodicity of water demands, and by extend water flows, to create time-varying water-quality models using online learning [92]. These approaches omit the information provided by hydraulic sensors, which may lead to significant estimation errors. In fact, the variability of water flows is the greatest source of uncertainty in water-quality models [60]. The most common approach to deal with modeling uncertainty due to water flows is using them as an input to the water-quality model, by assuming that the hydraulics are constant (steady-state) and known over a period of time called the hydraulic step [74]. The hydraulic-state estimate (flows and pressures) of the network during this period is obtained by combining measurements from hydraulic sensors and a suitable hydraulic-state estimator [93]. However, these estimates may include significant errors due to small number of available sensor measurements and hydraulic-model uncertainty [42]. During recent years, research on hydraulic-state estimation in WDN has developed more advanced methodologies which are able to quantify the hydraulic-state estimation error [38]. However a statistical characterization of this error is difficult to be calculated accuretely, so some researches have suggested the best way to represent it is by error bounds [40]. This is because in many cases, providing a range of values for each state is more useful to an operator than providing estimates with no indication of their proximity to the true state value.

Chlorine reaction dynamics in WDN, in most of the cases, are not known; as a result, empirical models (i.e., models created from laboratory and field experiments) are typically considered [94, 95]. A common assumption in water research literature is that chlorine dynamics are first-order linear, with some studies contacted in real networks showing this to be a good approximation [96]. Other studies are in favor of using more complex dynamics such as a second order model [97], as well as the use of differential-algebraic equations to model fast reactions [98]. In the first-order model, chlorine reaction dynamics depend on the bulk-water reaction coefficient (water source), the wall reaction coefficient (pipe material) and the mass transfer coefficient (chlorine transfer from bulk water to pipe walls) [99]. An estimate of wall and mass transfer coefficients can be made available given pipe material and age [100]. The bulk reaction rate dominates the decay dynamics, except in the case of highly reactive metallic pipes [96]. Bulk reaction rates are usually unknown and may vary due to the use of water originating from different sources (e.g. combining treated water from dams and desalination plants in different ratios) or fluctuations in temperature [96].

The water-quality modeling results in a set of hyperbolic partial differential equations (PDEs). In general, Eulerian or Lagrangian numerical methods can be used to find an approximate solution to the PDEs [91]. The Eulerian methods divide each link of the network into a number of equally sized segments, and transport water between segments at fixed intervals of time. The Lagrangian methods track the position of variable-sized segments of water in each link, computing new conditions at either fixed time intervals or at times when a new segment reaches the downstream node of a link. The later was adopted and used by the EPANET software, which uses a distributed model of water-quality, efficient at calculating output information at all network locations over time [74]. Another approach, belonging to the Lagrangian family, is the so-called Particle-Backtracking Algorithm (PBA) [101] which tracks individual water parcels from specific nodes (output) to chlorine injection nodes (input) backwards in time. This method has the advantage of retaining input-output relationship information, such as the time delay of injected chlorine reaching a sensor node. This advantage has been proven useful in calibrating pipe wall reaction coefficients, due to its ability to calculate the flow paths between input and output and creating an input-output model [102].

A model intended for water-quality monitoring as well as contamination detection should be able to quantify the estimation errors which arise from the combined hydraulic and water-quality parameter uncertainty. The generation of tight adaptive thresholds on the water-quality states can then be used for fault diagnosis and contamination detection [103]. One approach to do this is proposed in [9], where chlorine concentration bounds are calculated through Monte-Carlo Simulations (MCS) using the Lagrangian model of EPANET software and used as detection thresholds. The uncertain water demands and chlorine reaction rates are varied randomly within pre-defined bounds in each simulation. This is basically a simulation-oriented approach to set-membership state-estimation, in which uncertainty is modeled as an additive bounded error of which only the bounds are known. The result is a set-bounded state estimate, e.g., a state defined by an interval instead of a single value [104].

Powerful methodologies in set-bounded state estimation have been developed in the literature which can be applied on nonlinear discrete-time systems with unknown-but-bounded uncertainties [105]. However, the distributed nature of water-quality models in time and space make it challenging for generic set-membership estimation methodologies to be applied in practice. A notable application to water-quality is made in [55, 106]. Here, a distributed interval-model of water-quality is formulated following the Lagrangian modeling approach, based on known-bounded uncertainties on water flows and water-quality parameters. The researchers go a step further by proposing an interval observer structure for chlorine residual estimation. However, a drawback of this modeling approach is the increased interval-model complexity when the network size increases. Moreover, the switching nature of the model due to water flow reversal does not allow for sufficient stability conditions for the observer design, thus the assumption is made that flows are positive. Another key assumption is that the bulk reaction rate bounds are known, even though they may vary significantly if water from different sources is used. The proposed approach cannot facilitate real-time bulk reaction rate estimation as it does not consider time-varying parameters. As a result the uncertainty cannot be reduced, and the generated bounds are conservative, which increases the possibility of false negatives.

The above drawbacks could be mitigated by following an alternative modeling approach, such as the input-output model. This approach allows for the calculation of the state only at the output locations, thus reducing the complexity when network size increases. Additionally, due to the fact that path information can be retained, it can be used for parameter learning to reduce uncertainty. An additional benefit is that bulk reaction rate of individual water parcels can be tracked, thus allowing for variable bulk reaction rates in the network, which is useful when the system is supplied by multiple sources. However, an input-output model which considers water-quality-parameter and flow uncertainty has not yet been studied in the literature.

2.4 Contamination detection

A common practice for water utilities for water-quality monitoring and contamination detection is to manually take samples from certain location in the distribution network. This practice is not effective in the case of a contamination event since many customers will be affected before such event is confirmed. The recent approach of employing on-line water quality sensors enables the real-time monitoring of water quality. These sensors monitor generic water quality parameters such as chlorine concentration, oxidation reduction potential, total organic carbon, turbidity, conductivity and pH [107], since many of these parameters are affected in the case a contaminant enters the water. The location of the sensors is selected after solving the optimal sensor placement problem, a subject which motivated significant research in the past years [108, 89, 88, 109]. The methodology of fixed sensor locations for contamination detection can be improved, in terms of detectability, by choosing optimal sensor locations based on hydraulic and topological analysis of the system. Since the resulted set of sensor locations applies minimal to no change on the network topology or hydraulics, it is regarded as a Passive Contamination Detection (PCD) scheme, in which no manipulation on the systems' original (pre-setup) operation condition is being made. As opposed to the

PCD scheme described above, an Active Fault Detection (AFD) scheme considers deliberate reconfiguration of system components and hydraulics to achieve better detectability. This concept of AFD has been mainly studied in the past two decades in the fieled of electrical engineering [110, 111]. The methodology uses an alternative input procedure in known times to improve the detection ability of a given sensor set [112]. The field of AFD specifically for contamination detection in WDS lacks previous major researches.

The problem of contamination detection has been approached in various ways. Single-type or multiple-type measurements can be used for detection of contaminants, from one or more locations of the network. The measurements are then analyzed statistically without the need for a model of the water network (model-free), or are compared to calculated values from a network model (model-based). An example of a single-location multi-type measurement approach is given in [113], where each parameter was compared to its 3σ bounds. Other methods such as control charts and Kalman filters have also been proposed [114]. When multi-type sensors are available in more than one location, contamination can be detected by computing distance metrics [115], or by time series analysis of the measurements as performed in the CANARY event detection tool provided by the US Environmental Protection Agency [116, 117]. In the latter, an adaptive threshold is formed on the standard deviation of a moving window of measurements. In [118] a Bayesian Belief Network approach was presented as a method to infer the probability of contamination. In more recent work, the contamination event probability was calculated using artificial neural networks to model water quality and produce an estimation error which was then compared to a threshold [119]. The probability was calculated by utilizing a sequential Bayesian rule. Furthermore, in [120], this approach was extended to consider dynamic thresholds computed with respect to the measurements from a moving window. In [121], an un-supervised approach was proposed which uses a minimum-volume ellipsoid to define the region of normal measurements. A heuristic metric then determines the probability that an abnormality has occurred.

The use of model-based approaches for contamination event detection require the knowledge of multiple chemical reaction dynamics. The underlying assumption is that certain contaminants when injected in the water will affect key monitored parameters in a specific way [29]. For example, a bacterial toxin may decrease the concentration of free chlorine, decrease the ORP and increase the conductivity of the water. The use of chlorine as a disinfectant by many water utilities results in chlorine sensors being widely used in water distribution networks for monitoring chlorine residuals. The use of chlorine sensors specifically for contamination event detection was proposed in [122], and it can be a low cost solution for contamination detection. The actual chlorine reaction dynamics in most of the cases are not known and as a result, empirical models are utilized [94]. In addition, models describing chlorine reactions with contaminants (such as sodium arsenite and organophosphate) have been proposed [123]. In [28], the EPANET-MSX software [98] was used to simulate the chlorine response to the injection of certain biological agents in a water distribution benchmark network. Furthermore, in [124], chlorine and contaminant reaction models have been considered in a real-time event adaptive detection, identification and warning methodology, to detect and classify the contaminant.

Water quality models used for estimating chlorine residuals often do not give accurate results due to their time-varying nature. The greatest source of uncertainty is introduced by the variability of water demands [60]. In [55], a chlorine concentration interval estimator was implemented which calculates an interval of values instead of a single value for chlorine concentration, by considering the uncertainty in water flows. Hydraulic and quality control actions also contribute to the fluctuation of disinfectant residuals and must be considered [125]. The use of fixed detection thresholds for chlorine residuals can make the event detection insensitive to small or incipient contamination events or can trigger false alarms in the case of abrupt hydraulic actions. In [126], a method for computing bounds of the expected chlorine concentration at different chlorine sensing locations is presented which considers the known chlorine input injection signals and also takes into account the uncertainties in hydraulic dynamics. The bounds are calculated using multiple Monte-Carlo simulations which run in parallel to the real system.

Chapter 3

Water distribution systems hydraulics modeling

This chapter formulates the hydraulic model of a WDS considering bounded parameter and measurement uncertainty. First the hydraulic equations, derived from the laws of conservation of mass and energy in these systems, are formulated. The measurements set needed to obtain observability is then defined. Uncertainty is modeled using the bounds on uncertain parameters to represent them as intervals. Measurement uncertainty arises due to the need of using artificial demand estimates called pseudo-measurements which are highly uncertain and without a statistical characterization of the mismatch to the real value. Modeling uncertainty is also considered due to uncertain pipe parameters such as the roughness coefficients. Moreover, a state-space formulation of hydraulics is proposed for the cases when tanks exist in the system.

Notation: Matrices are denoted with capital bold letters, vectors with lower bold letters, and scalars by italic letters. Sets and graphs are denoted by calligraphic letters. Uncertain parameters are represented by a continuous interval of values defined by a lower and upper bound. Intervals are accompanied by a tilde and defined as follows: Let $\tilde{\boldsymbol{v}} = [\boldsymbol{v}^l, \boldsymbol{v}^u]$ be a closed interval vector, where $\boldsymbol{v}^l$ is the lower bound vector and $\boldsymbol{v}^u$ is the upper bound vector, such that: $\tilde{\boldsymbol{v}} = \{\boldsymbol{v} \in \mathbb{R}^n : v_i^l \leq v_i \leq v_i^u, \forall\, i = \{1, .., n\}\}$, and n is the size of the vector.

3.1 Hydraulics modeling

The topology of a WDN is modeled by a directed graph denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Let $\mathcal{N} = \{1, \cdots, n_n\}$ be the set of all nodes, where $|\mathcal{N}| = n_n$ is the total number of nodes.

These represent junctions of pipes, consumer water demand locations, reservoirs and tanks. The unknown quantity associated with nodes is the *hydraulic head*, indicated by h_j. This is a measure of energy consisting of a component proportional to the pressure p_j at node j, and of the node elevation with respect to a geodesic reference z_j [127], such that:

$$h_j = p_j + z_j. \tag{3.1}$$

Each node j is associated with a water consumer demand at the node location, denoted by $q_{ext,j}$. Certain nodes, such as reservoirs, have a known head and zero demand. We define the set of nodes with known head $\mathcal{N}_h$, where $|\mathcal{N}_h| = n_h$ is the number of nodes with known head. The set of nodes with unknown head is defined as $\mathcal{N}_u$, where $|\mathcal{N}_u| = n_u$ is the number of nodes with unknown head and $\mathcal{N} = \mathcal{N}_u \cup \mathcal{N}_h$.

Let $\mathcal{L} = \{1, \cdots, n_l\}$ be the set of links, where $|\mathcal{L}| = n_l$ is the total number of links. These represent network pipes, water pumps and pipe valves, with the last two being the main hydraulic control elements in a water network. We define the set of links that represent pipes as $\mathcal{L}_p = \left\{1, \cdots, n_p\right\}$, where $\left|\mathcal{L}_p\right| = n_p$ is the total number of pipes. We also define the set of links that represent pumps as $\mathcal{L}_{pu} = \left\{n_p + 1, \cdots, n_l\right\}$, where $\left|\mathcal{L}_{pu}\right| = n_{pu}$ is the total number of pumps. The unknown quantity associated with a link i is the *water flow*, indicated by q_i.

Each component of the WDS graph is associated with certain parameters that describe different physical properties of the network and are used in hydraulic-state estimation. For pipes, these are the pipe roughness coefficients, length and diameter, while for nodes these are the node elevations. In this work we consider these parameters time-invariant and denote them using a single vector $\boldsymbol{\theta}$. In extension, let $\mathcal{M}(\boldsymbol{\theta}; \mathcal{G})$ be a complete model of a WDS which associates the network graph with each network parameter.

The complete hydraulic-state of a WDS, comprised of the water flows in pipes and hydraulic heads at nodes, is indicated by $\boldsymbol{x} = [\boldsymbol{q}^\top \ \boldsymbol{h}^\top]^\top \in \mathbb{R}^{n_l + n_u}$. The state can be calculated using a hydraulic model which is a set of equations constructed from the laws of *conservation of energy* and *conservation of mass* in the network. In this work the *pipe formulation* of these equations is used [128], which has been shown to be robust in computer simulations [74].

3.1.1 Conservation of energy equations

Energy in WDN is associated with the head at nodes and when water flows through a network link i which connects two nodes, a flow dependent head function $f_i(q_i)$ describes the change in head. In the case of pipes, energy is dissipated due to friction of water flowing through the pipe, resulting in head-loss between two connected nodes. Head-loss depends on the water flow through the pipe but also on pipe parameters. Each pipe $i \in \mathcal{L}_p$ is characterized by pipe length l_i, pipe diameter d_i and pipe roughness coefficient c_i. All these quantities are used in the empirical Hazen-Williams (H-W) formula [127] to calculate head-loss. The effect of pipe parameters in this formula is aggregated in the H-W resistance coefficient r_i of each pipe, which is a function $f^r_{HW} : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \mapsto \mathbb{R}^+$ of pipe parameters, defined as: $r_i = f^r_{HW}(c_i, d_i, l_i)$. The head-loss across pipe i is then calculated using the H-W formula as follows:

$$f_i(q_i) \triangleq r_i \left|q_i\right|^{(\nu-1)} q_i\,,\ i \in \mathcal{L}_p, \tag{3.2}$$

where ν is a constant exponent associated with the H-W formula and q_i is the water flow in pipe i.

Another example of a network element are pumps which are characterized by a head-flow curve. This curve is used to relate the flow through the pump to the head-gain across the pump, according to each pump specifications. This is given by:

$$f_i(q_i) \triangleq -(w_1 - w_2 q_i^{w_3})\,,\ i \in \mathcal{L}_{pu}, \tag{3.3}$$

where $w_1, w_2, w_3 \in \mathbb{R}$ are coefficients of the pump head-flow curve, while the minus sign indicates that in the case of pumps there is head-gain instead of head-loss. Other network elements with a head-flow function which may exist in WDS and are not described here include valves such as Pressure Reduction Valves (PRV), Flow Control Valves (FCV) and General Purpose Valves (GPV).

The conservation of energy law when using the pipe formulation dictates that the head-loss across a link $i \in \mathcal{L}$ is equal to the head difference between the two nodes connected by the link. The energy equations for all the network links, considering elements modeled by (6.2) and (3.3), can be written as follows:

$$f_i(q_i) + \sum_{j \in \mathcal{N}_u} \left(B_{ij}\, h_j\right) = h_{ext,i}, \quad i \in \mathcal{L}, \tag{3.4}$$

where:

- h_j is the unknown head of node $j \in \mathcal{N}_u$.
- $B \in \mathbb{R}^{n_l \times n_u}$ is the incidence flow matrix, indicating the connectivity of nodes with links. Element $B_{ij} = +1$ if the direction of link i enters node j; element $B_{ij} = -1$ if the direction of link i leaves from node j; otherwise $B_{ij} = 0$. Nodes with known head are excluded from this matrix.

- $h_{ext,i}$ is the sum of known (measured) heads that appear in each equation $i \in \mathcal{L}$. In vector notation, the known head vector is given by $\boldsymbol{h}_{ext} = B_0 \boldsymbol{h}_0 \in \mathbb{R}^{n_l}$, with $\boldsymbol{h}_0 \in \mathbb{R}^{n_h}$ being the vector of known heads and $B_0 \in \{-1, 0, 1\}^{n_l \times n_h}$ being the mapping of known heads to each equation.

3.1.2 Conservation of mass equations

The conservation of mass law for a node $j \in \mathcal{N}_u$ is similar to Kirchhoff's current law in electric circuit analysis and can be summarized as follows: the sum of branch water flows from pipes incident to a node j must be equal to the node's external water demand $q_{ext,j}$.

A demand-driven modeling approach assumes that the demand at each node is independent of the pressure at that node. However, this analysis is not valid when power outages, fire fighting, pipe breaks or temporarily closed portions of a WDN lead to pressure-deficit conditions. In those cases, the consumers do not receive the requested demand, thus the modeling of demand is no longer valid and a pressure-dependent demand modeling is recommended. The pressure-demand relationship can be modeled by multiplying the user requested demand $q_{ext,j}$ at node j by the pressure depended function $f_{ext,j}(h_j) \in [0, 1]$, which is given by [129, 130, 131]:

$$f_{ext,j}\left(h_j\right) \triangleq \begin{cases} 0 & h_j \leqslant h_{\min,j} \\ \left(\frac{h_j - h_{\min,j}}{h_{req,j} - h_{\min,j}}\right)^{0.5} & h_{\min,j} \leqslant h_j \leqslant h_{req,j} \\ 1 & h_j \geqslant h_{req,j} \end{cases} . \tag{3.5}$$

In (3.5), $h_{req,j}$ is the head above which the consumer can receive the requested demand $q_{ext,j}$ (depends on node elevation), $h_{min,j}$ is the minimum desired head at node j (depends on node elevation) below which the consumer does not receive any water.

Background leakage flows are also present in real WDN and are modeled as an added demand component at nodes. Leakage flows are pressure-depended and are modeled similarly to pressure-driven demands as follows [132]:

$$q_{leak,j}(h_j) \triangleq \begin{cases} c_j (h_j - z_j)^{\alpha_j}, & h_j - z_j > 0 \\ 0 & h_j - z_j \leqslant 0 \end{cases} \tag{3.6}$$

where z_j is the elevation of node j, and c_j and α_j are leakage parameters depending on pipe deterioration and material.

The conservation of mass equations in the most general case, are dynamic due to the presence of tanks in the network. The generic tank equation is given by:

$$\alpha_j \dot{h}_j(t) = q_{in,j} - q_{out,j}, \tag{3.7}$$

where α_j is a known constant related to the tank dimensions of node j, q_{in} is the tank inflow and q_{out} the outflow. The general case of the conservation of mass equations is given as follows:

$$\alpha_j \dot{h}_j(t) = \sum_{i\in\mathcal{L}} \left(B_{ij}^\top q_i(t)\right) - q_{ext,j}(t)\, f_{ext,j}(h_j(t)) - q_{leak}(h_j(t)), \quad j \in \mathcal{N}_u. \tag{3.8}$$

A simplified version of the mass conservation equations can be formulated by assuming a demand-driven modeling approach. This approach assumes that the network operates in pressure-sufficient conditions and thus demand at each node is independent of the node pressure. Background leakages can be incorporated in the demand variables $q_{ext,j}$, since these are essentially an additional demand at the node. Moreover, when the system does not contain tanks or when the tank levels are assumed known, the mass balance equations become algebraic thus the time notation is omitted. The simplified mass conservation equations are given by:

$$\sum_{i\in\mathcal{L}} \left(B_{ij}^\top q_i\right) = q_{ext,j}, \quad j \in \mathcal{N}_u. \tag{3.9}$$

Equations (6.1) and (6.3) can be used to define the network complete hydraulic state, which are the water flows in pipes and hydraulic heads at nodes, indicated by $\boldsymbol{x} = [\boldsymbol{q}^\top \; \boldsymbol{h}^\top]^\top$.

3.1.3 Measurement uncertainty

In this work we follow the approach of modeling uncertain quantities using intervals, which is equivalent to the actual value of this quantity having a uniform probability distribution. For notational convenience, we adopt the convention of denoting intervals with a tilde. Let $\tilde{\boldsymbol{v}} = [\boldsymbol{v}^l, \boldsymbol{v}^u]$ be a closed interval vector, where $\boldsymbol{v}^l$ is the lower bound vector and $\boldsymbol{v}^u$ is the upper bound vector, such that: $\tilde{\boldsymbol{v}} = \{\tilde{v} \in \mathbb{R}^n : v_i^l \leq v_i \leq v_i^u, \forall\, i = \{1,..,n\}\}$, and n is the size of the vector. Note that calculations performed in equations containing intervals require the use of interval arithmetic [133, 134, 135].

Sensor measurements in WDS arrive at discrete time intervals Δt_1, typically a few minutes, corresponding to measurement time-steps indicated by k. In the case of a typical District Metered Area (DMA) in a WDS, flow measurements $\hat{q}_j(k)$ are available at the inlet, where $j \in \mathcal{L}_{in} \subset \mathcal{L}$ and $|\mathcal{L}_{in}| = n_i$ is the number of inlets. Additionally, a number of pressure sensors are installed inside the DMA, giving pressure measurements indicated by $\hat{p}_i(k), i \in \mathcal{N}_s \subset \mathcal{N}$, where $|\mathcal{N}_s| = n_s$ is the number of pressure sensors. The complete measurement vector is indicated by $\boldsymbol{y}(k) \in \mathbb{R}^{n_i+n_s}$. Sensor measurements include measurement noise, which is assumed bounded, such that the true value of measured state relates to the measurements as follows:

$$\boldsymbol{y}(k) \triangleq \begin{bmatrix} \hat{\boldsymbol{q}}(k) \\ \hat{\boldsymbol{p}}(k) \end{bmatrix} = \begin{bmatrix} C_q & C_h \end{bmatrix} \begin{bmatrix} \boldsymbol{q}(k) \\ \boldsymbol{h}(k) - \boldsymbol{z} \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_q(k) \\ \boldsymbol{v}_h(k) \end{bmatrix}, \tag{3.10}$$

where C_q and C_h identify measured flow and head states and $\boldsymbol{v}_q \in \tilde{\boldsymbol{v}}_q$, $\boldsymbol{v}_p \in \tilde{\boldsymbol{v}}_p$ are the noise vectors for flow and pressure devices respectively, which are assumed bounded by known bounds.

WDS have typically a low number of sensors installed, i.e., $(n_s + n_i) << n_n$, which makes the system of equations defined by (6.1) and (6.3) under-determined. According to the analysis in [23], at least n_n sensors in an observable configuration are needed to guarantee the topological observability of the system. A possible observable configuration is when water demands are measured at all nodes and at least one additional head measurement is available. Other sensor configurations are possible, given that they satisfy the topological observability condition, which can be checked using the algorithm in [23]. Consumer demands are modeled as an uncontrolled input to the system, thus demand measurements are not represented in the measurement vector of (6.6).

In this work we assume that demand estimates called *pseudo-measurements* are used to complement missing measurements and make the problem over-determined [42]. Water demands exhibit some regularity based on the consumer habits and geographical distribution, thus demand estimates for each time step k can be created using off-line information such as population densities, building areas and the consumption patterns of typical consumers [136]. These estimates are then further improved by combining them with available low-resolution measurements, such as quarterly billing data. They can also be a function of the current inflow, as shown in [50]. However, these pseudo-measurements are highly uncertain with no available statistical characterization. The most accurate depiction of demand pseudo-measurements is in the form of error bounds [40]. The characteristics of, but also the need for, pseudo-measurements in state-estimation is the main reason why uncertainties are modeled in this work using intervals. The uncertain demand pseudo-measurements are given for each discrete time step k as follows:

$$\tilde{\boldsymbol{q}}_{ext}(k) \triangleq \left[\boldsymbol{q}^{l}_{ext}(k),\ \boldsymbol{q}^{u}_{ext}(k)\right]. \tag{3.11}$$

It is also possible for Automatic Meter Reading (AMR) devices to be installed at some demand nodes. For the purpose of modeling, each AMR is represented as a single node in the network graph. These provide the consumption of consumers at predefined time steps

Δt_2 (e.g. 2 hours) or at each time instant that a specific volume of water is consumed. Typically AMRs report less frequently than pressure or flow sensors installed in the system, i.e., $\Delta t_2 > \Delta t_1$. In order to maintain a common time step when solving the hydraulic equations, the smallest measurement time step Δt_1 of the pressure measurements is used. An average estimate of the demand flow $\hat{q}_{ext,j}(k)$ at AMR node j can be calculated based on the volume consumption. These demand estimates contain considerable uncertainty arisen from the larger time step and loss of information during the conversion of measurement units. An upper bound on this uncertainty can be defined for each time step k, such that the AMR measurements have the same representation as pseudo-measurements, given by (3.11).

3.1.4 Model uncertainty

The topology of a WDS is assumed known, i.e., the graph $\mathcal{G}$ represents the actual system structure. As a practical note, a calibration pre-step of network topology is recommended [137], as it will identify mismatches in valve status (open/closed) which is the main uncertainty regarding network topology [42].

We also consider the uncertainty on the head-loss function $f_i\,(q_i)$. When this function contains uncertain parameters, these will be modelled as intervals defined by a lower and upper bound, and the head function will be indicated in bold as $\boldsymbol{f}_i\,(q_i)$. Uncertainty in pipe parameters is included in the uncertain H-W coefficients r_i. These are calculated using uncertain pipe parameters, which are the roughness coefficients c_i, diameter d_i and length l_i. For a certain pipe i, the uncertain H-W coefficient is given by: $\tilde{r}_i = \left[r_i^l,\ r_i^u\right]$. Uncertainty in model parameters is modeled as an uncertain head-loss function $\tilde{f}_i\,(q_i)$, where the tilde indicates that this function contains uncertain parameters:

$$\tilde{f}_i\,(q_i) \triangleq \tilde{r}_i\,|q_i|^{\nu-1}q_i, \quad i \in \mathcal{L}_p. \tag{3.12}$$

Similarly, for i corresponding to a pump with an uncertain pump curve, (3.3) becomes:

$$\tilde{f}_i(q_i) \triangleq -(\tilde{w}_1 - \tilde{w}_2 q_i^{\tilde{w}_3}), \quad i \in \mathcal{L}_{pu}. \tag{3.13}$$

Note that this approach of modeling uncertainties can be applied to the head-loss function of any element of the system modeled by a link. Model parameter bounds are assumed known and constant, because these parameters vary slowly over time.

3.1.5 Algebraic hydraulic equations with uncertainty

The problem of solving the algebraic hydraulic equations of a WDS when these contain uncertainty in the form of intervals, is reduced to finding the set of all point solutions for the state $\boldsymbol{x}(k) = [\boldsymbol{q}(k)^\top \ \boldsymbol{h}(k)^\top]^\top$, that satisfy the following systems of equations:

$$\tilde{f}_i\left(q_i(k)\right) + \sum_{j\in\mathcal{N}_u} \Big(B_{ij}\, h_j(k)\Big) = \tilde{h}_{ext,i}, \qquad i \in \mathcal{L} \qquad (3.14\alpha')$$

$$\sum_{i\in\mathcal{L}} \Big(B_{ij}^\top\, q_i(k)\Big) = \tilde{q}_{ext,j}(k), \qquad j \in \mathcal{N}_u \qquad (3.14\beta')$$

3.2 State-space hydraulic model

When tanks exist in the system, the hydraulic-equations contain dynamic states corresponding to tank levels, as given by (3.8). In this case, the hydraulic equations of a water network can be written in state-space form as follows:

$$E\underbrace{\begin{bmatrix} \dot{\boldsymbol{h}}(t) \\ \dot{\boldsymbol{q}}(t) \end{bmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{bmatrix} 0 & B^\top \\ B & F\left(\boldsymbol{q}(t)\right) \end{bmatrix}}_{A(\boldsymbol{x}(t))}\underbrace{\begin{bmatrix} \boldsymbol{h}(t) \\ \boldsymbol{q}(t) \end{bmatrix}}_{\boldsymbol{x}(t)} + K\boldsymbol{u}(t) + G\underbrace{\begin{bmatrix} -\boldsymbol{q}_{ext}(t) \\ \boldsymbol{h}_{ext} \end{bmatrix}}_{\boldsymbol{d}(t)}, \qquad (3.15)$$

where:

- $E \in \mathbb{R}^{n\times n}$ identifies the dynamic states corresponding to tanks levels and multiplies them with the tank base area,
- $B \in \mathbb{R}^{n_l\times n_u}$ is the network incidence matrix indicating the connectivity of nodes with links (nodes with known head, such as reservoirs, are excluded from this matrix),
- $F(\boldsymbol{q}(t)) : \mathbb{R}^{n_l} \mapsto \mathbb{R}^{n_l\times n_l}$, contains the nonlinear functions $f_i(q_i), i \in \mathcal{L}$ at its diagonal, which calculate the head-loss across pipe i using pipe parameters $\boldsymbol{\theta}$ and according to the Hazen-Williams empirical formula [127],
- $\boldsymbol{q}_{ext}(t) \in \mathbb{R}^{n_u}$ are the external water demands at nodes,
- $\boldsymbol{h}_{ext} \in \mathbb{R}^{n_n-n_u}$ is a vector of known heads (such as the head of reservoirs),

The vector $\boldsymbol{d}(t)$ contains the known node heads and the external water demands, which drive the system dynamics. These are modeled as an uncontrolled, known input to the system. Matrix $G \in \mathbb{R}^{n\times n}$ identifies the demand and known head components in each equation.

The control signals $\boldsymbol{u}(t) \in \mathbb{R}^{n_c}$ in (3.15) represent the ways an operator can interfere with the network operation. The matrix $K \in \mathbb{R}^{n\times n_c}$ identifies the control input in each equation. The modeling of control inputs here corresponds to changing the settings of pumps that exist in the network by altering the pump head-flow curve [127]. Another actuator group in a WDS are hydraulic valves on pipes. When a valve closes it changes the topology of the network, modifying the incidence matrix B. Here we assume a time-invariant matrix B, so this kind of inputs are not modeled in this formulation.

3.2.1 Separation of dynamic and algebraic states

The matrix E in (3.15) is singular due to the presence of algebraic equations, therefore it is necessary to transform the system into a non-singular form. This can be achieved by exploiting specific results from index-one singular systems, as shown in [138, 139], to divide the state vector $\boldsymbol{x}$ into two subsets: the dynamic states indicated by $\boldsymbol{x}_1 \in \mathbb{R}^{n_t}$, and the algebraic states indicated by $\boldsymbol{x}_2 \in \mathbb{R}^{(n_u+n_l-n_t)}$, where n_t is the number of tanks (equal to the number of dynamic states of the system). To achieve this, the dynamic states are rearranged and separated from the algebraic states, using the following transformation:

$$E_t = TET^{-1},\ A_t = TA(\boldsymbol{x})T^{-1},\ K_t = TK,\ G_t = TG, \tag{3.16}$$

where T is the permutation matrix that is formulated by permuting the rows of the identity matrix $I_{n \times n}$, so that the rows corresponding to tank head-states are moved to the top. It is noted that the tank states correspond only to linear equations, therefore the non-linearities are associated only with the algebraic equations. The transformed state-space equations can then be written as follows:

$$\underbrace{\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}}_{E_t} \underbrace{\begin{bmatrix} \dot{\boldsymbol{x}}_1(t) \\ \dot{\boldsymbol{x}}_2(t) \end{bmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22}(\boldsymbol{x}_2(t)) \end{bmatrix}}_{A_t(t)} \underbrace{\begin{bmatrix} \boldsymbol{x}_1(t) \\ \boldsymbol{x}_2(t) \end{bmatrix}}_{\boldsymbol{x}(t)} + \underbrace{\begin{bmatrix} K_1 \\ K_2 \end{bmatrix}}_{K_t} \boldsymbol{u}(t) + \underbrace{\begin{bmatrix} G_1 \\ G_2 \end{bmatrix}}_{G_t} \boldsymbol{d}(t), \tag{3.17}$$

where $E_{11}, A_{11} \in \mathbb{R}^{n_t \times n_t}$, while the rest of the matrices have the appropriate size defined by the division of the states. The dynamic equations and the algebraic equations can then be written separately as follows:

$$\dot{\boldsymbol{x}}_1(t) = \underbrace{E_{11}^{-1}A_{11}}_{A_{x_1}} \boldsymbol{x}_1(t) + \underbrace{E_{11}^{-1}A_{12}}_{A_{x_2}} \boldsymbol{x}_2(t) + \underbrace{E_{11}^{-1}K_1}_{K_{x_1}} \boldsymbol{u}(t) + \underbrace{E_{11}^{-1}G_1}_{G_{x_1}} \boldsymbol{d}(t), \tag{3.18α′}$$

$$0 = A_{21}\boldsymbol{x}_1(t) + A_{22}(\boldsymbol{x}_2(t))\boldsymbol{x}_2(t) + K_2\boldsymbol{u}(t) + G_2\boldsymbol{d}(t). \tag{3.18β′}$$

Notice that if we consider the algebraic states $\boldsymbol{x}_2$ as an input, then the dynamic equations of (3.18α′) are linear. The algebraic equations of (3.18β′) however are nonlinear and cannot be solved analytically with respect to $\boldsymbol{x}_2$.

3.2.2 Uncertainty, measurements and discretization

Demand uncertainty is considered by replacing the uncontrolled input vector with the interval vector $\tilde{\boldsymbol{d}}(k)$. Model uncertainty is also present in the form of uncertain pipe parameters $\boldsymbol{\theta}$. We assume that pipe parameters are uncertain with a known uncertainty bound and represented by $\tilde{\boldsymbol{\theta}}$. These uncertain parameters are included in the matrix A_{22} which now contains interval parameters, thus it is indicated by $\tilde{A}_{22}$.

Measurements in the system include flow and pressure measurements as defined in (6.6), and additionally tank level measurements. The system measurements can be written as:

$$\boldsymbol{y}(k) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1(k) \\ \boldsymbol{x}_2(k) \end{bmatrix} + \tilde{\boldsymbol{v}}(k), \tag{3.19}$$

where $\boldsymbol{y}(k) \in \mathbb{R}^{n_s}$ is the measurement vector, n_s is the total number of sensors, and $\tilde{\boldsymbol{v}}(k)$ is the measurement noise vector which is assumed bounded and represented by an interval. Moreover, we can distinguish between measurements by defining $\boldsymbol{y}_1(k) = C_1\boldsymbol{x}_1(k)$ as the tank level measurements and $\boldsymbol{y}_2(k) = C_2\boldsymbol{x}_2(k)$ as the sensor measurements defined by (6.6).

The state-space equations of (3.18) can then be discretized based on the measurement time step Δt_1. Discretized matrices are indicated by the superscript d. The matrices in the algebraic equations (3.18β′) remain the same after the discretization procedure. The discrete-time state-space equations of a WDS considering bounded uncertainties are given by:

$$\boldsymbol{x}_1(k+1) = A^d_{x_1}\boldsymbol{x}_1(k) + A^d_{x_2}\boldsymbol{x}_2(k) + K^d_{x_1}\boldsymbol{u}(k) + G^d_{x_1}\tilde{\boldsymbol{d}}(k) \tag{3.20α′}$$

$$0 = A_{21}\boldsymbol{x}_1(k) + \tilde{A}_{22}(\boldsymbol{x}_2(k))\boldsymbol{x}_2(k) + K_2\boldsymbol{u}(k) + G_2\tilde{\boldsymbol{d}}(k). \tag{3.20β′}$$

Note that the algebraic equations of (3.20β′) are identical to the algebraic equations derived in (3.14).

Chapter 4

Hydraulic interval-state estimation

In this chapter we propose a new methodology for hydraulic interval-state estimation for WDS that considers the combined effect of bounded measurement and modeling uncertainties. The proposed *Iterative Hydraulic Interval State Estimation* (IHISE) methodology first calculates the bounds on state estimates using the nonlinear form of the network algebraic equations, by also modeling pressure-dependent demands. The nonlinear modeling guarantees that when accurate uncertainty bounds are provided, the bounds on state estimates will include the true system state. This is achieved using bounding linearization, a technique which restricts the nonlinearities within a convex set, thus converting the hydraulic equations in a form where the minimum and maximum of each state can be found using linear optimization. Then, an iterative procedure is followed to minimize the distance between upper and lower state bounds, by improving the bounding linearization at each step and converging to the tightest possible bounds. In order to facilitate system dynamics, a state-space model is formed and the dynamic states are separated from the algebraic states. An overall framework for interval state estimation is then developed which uses the IHISE algorithm with an interval observer for the dynamic states. Simulation results demonstrate that the proposed methodology produces tight state bounds that can replace Monte-Carlo simulations.

This remainder of this chapter is organized as follows: Section “Introduction” describes the problem to be solved in the context of recent literature. Section “Iterative Hydraulic Interval State Estimation” presents a methodology to solve this problem based on the Iterative Hydraulic Interval State Estimation (IHISE) algorithm. In Section “Case Studies” this methodology is applied on different benchmark water networks and its performance is assessed.

4.1 Introduction

Problem (3.14) is a Nonlinear Interval System of Equations (NISE) and the set of solutions for $\boldsymbol{x}$ that satisfy (3.14) may have a complex form that needs to be described with nonlinear functions. This is why, in the literature, 'interval solutions' are most often considered, with the aim of finding the Interval Hull (IH) solution, i.e., the smallest interval vector $\tilde{\boldsymbol{x}}$ containing all solutions for the NISE. Finding the IH solution to general NISE is a challenging problem; even for the general Linear Interval System of Equations (LISE), finding the IH is an NP-hard problem. For this reason there are several solutions proposed in the literature that approximate the IH, either with an Outer Interval (OI) solution, which is any interval vector enclosing the IH solution, or with an INner Interval (INI) solution, which is any interval vector that is a subset of the IH solution. Most approaches deal with the problem of finding an OI solution, while the INI solution is used as a measure of the overestimation of the solution [140, 141]. An example of a method for finding the INI solution to a NISE are Monte-Carlo Simulations, where solutions are calculated by randomly generating and evaluating a large number of non-interval equations with parameters within the defined intervals [9]. The set of solutions is always a subset of the IH solution.

The literature on finding an OI solution to NISE is limited, but some approaches have been proposed, such as [142], which uses affine arithmetic to represent the equations and interval linearization to deal with the nonlinearities. However, this approach does not consider interval multiplicative terms in the nonlinear functions, thus cannot be applied to (3.14). The solution to NISE can also be approached using optimization, as in [143] where the task of solving nonlinear interval number programming problems was investigated. This method, however, does not ensure that the solution is an OI, thus it is not suitable for use with methodologies such as fault detection which require outer bounds on states.

Good approaches in the literature that provide tight OI solutions exist for Linear Interval Systems of Equations (LISE) and are mainly divided in two categories. The first uses interval arithmetic [133, 135] to find the solution. Due to the fact that when using interval arithmetic to solve LISE, the solution interval is inherently an overestimation, iterative methods are used to approximate the IH solution, such as the Gauss Elimination method, LU decomposition method and the iterative Jacobi method [144]. The second category formulates LISE as an Interval Linear Programming problem, where intervals can exist both in the objective function and in the constraints [145, 146]. This approach is promising since the formulation of the equations as a Linear Program provides the opportunity to add and manipulate constraints to the problem. Additionally, powerful software that solve Linear Programs efficiently exist, which reduce computation time.

4.2 Iterative Hydraulic Interval State Estimation

In this work we propose an iterative method for finding an OI solution of the NISE in (3.14), named *Iterative Hydraulic Interval State Estimation* (IHISE), which closely approximates the IH solution. This method deals with the nonlinearities in (3.14) using bounding linearization, which encloses the interval nonlinearities in a convex set and converts (3.14) into a system of linear inequalities. The resulting linear inequalities are then appropriately formulated into a Linear Program and new bounds on the state variables are calculated. An iterative procedure then approximates the IH solution of (3.14) by using the new bounds on the states to improve the bounding linearization. Initial bounds on state variables can be defined from physical properties of WDS.

The IHISE algorithm is comprised of five main steps:

1. Find initial bounds on the state variables using physical constraints of the system.
2. Use bounding linearization to bound the nonlinearities in a convex set.
3. Formulate Linear Programs (LPs) for each state using the resulting linear inequalities.
4. Solve one maximization and one minimization LP for each state to produce new upper and lower bounds.
5. Iteratively improve the OI solution of (3.14) by repeating steps 2 to 4 until convergence of bounds.

Next, the five steps of IHISE are described in detail.

4.2.1 Step 1: Initial bounds on state vector

The first step of the IHISE algorithm is to impose initial bounds on the state vector $\boldsymbol{x} = [\boldsymbol{q}^\top \ \boldsymbol{h}^\top]^\top$ which should be an OI solution of (3.14). The initial bounds on the unknown head vector $\boldsymbol{h}$ are given by the interval vector $\tilde{\boldsymbol{h}}^{(0)} = \left[\boldsymbol{h}^{l\,(0)},\ \boldsymbol{h}^{u\,(0)}\right]$ and are chosen by considering physical properties of the network. The minimum head vector $\boldsymbol{h}^{l\,(0)}$ is set equal to the elevation of each node and the maximum head vector $\boldsymbol{h}^{u\,(0)}$ is set equal to the sum of reservoir and pump heads, which is the maximum head that any node in the network can have.

The special structure (3.14α′), in which each equation contains only one flow state q_i, allows us to use the initial bounds on heads $\tilde{\boldsymbol{h}}^{(0)}$ to find the initial bounds on the flows. We rewrite (3.14α′) as follows:

$$\tilde{f}_i(q_i) = -\sum_{j\in\mathcal{N}_u}\left(B_{ij}\,h_j^{(0)}\right) + \tilde{h}_{ext,i} = \tilde{y}_i, \tag{4.1}$$

where the function $\tilde{f}_i(q_i)$ is given by (6.10) when $i \in \mathcal{L}_p$, and $\tilde{y}_i = \left[y_i^l, y_i^u\right]$ is a known interval derived from the known terms in (4.1) using interval analysis. The function $\tilde{f}_i(q_i)$ is *inclusion isotonic* [135] meaning that if $\tilde{q}_i^1 \subseteq \tilde{q}_i^2$ then $\tilde{f}(q_i^1) \subseteq \tilde{f}(q_i^2)$. This property enables the derivation of analytical bounds on the unknown pipe flows, by rearranging (4.1) with respect to q_i, $i \in \mathcal{L}_p$, as follows:

$$\left[q_i^{l\,(0)}, q_i^{u\,(0)}\right] = \begin{cases} \left[-\left(\frac{|y_i^u|}{r_i^l}\right)^{\frac{1}{\nu}}, -\left(\frac{|y_i^l|}{r_i^u}\right)^{\frac{1}{\nu}}\right], & y_i^l < 0, y_i^u < 0 \\ \left[-\left(\frac{|y_i^l|}{r_i^l}\right)^{\frac{1}{\nu}}, +\left(\frac{|y_i^u|}{r_i^l}\right)^{\frac{1}{\nu}}\right], & y_i^l < 0, y_i^u > 0 \\ \left[+\left(\frac{|y_i^l|}{r_i^u}\right)^{\frac{1}{\nu}}, +\left(\frac{|y_i^u|}{r_i^l}\right)^{\frac{1}{\nu}}\right], & y_i^l > 0, y_i^u > 0 \end{cases}. \tag{4.2}$$

In the case of pumps, $\tilde{f}_i(q_i)$ is given by (3.13). This function is not inclusion isotonic in its whole range, but this property holds in the special case when $q_i > 0$ or $q_i < 0$. This implies that the flow direction in pump links needs to be known *a priori*, which is a valid assumption, since pumps are physically restricted to move water in one direction. Assuming that the flow in pump links is always positive, the bounds on flows q_i, $i \in \mathcal{L}_{pu}$ can be found by rearranging (4.1) with respect to q_i, as follows:

$$q_i^{l\,(0)} = \min\left(\left(\frac{y_i^l - w_1}{w_2}\right)^{\frac{1}{w_3}}, \left(\frac{y_i^u - w_1}{w_2}\right)^{\frac{1}{w_3}}\right), \tag{4.3}$$

$$q_i^{u\,(0)} = \max\left(\left(\frac{y_i^l - w_1}{w_2}\right)^{\frac{1}{w_3}}, \left(\frac{y_i^u - w_1}{w_2}\right)^{\frac{1}{w_3}}\right). \tag{4.4}$$

The initial bounds on the flow state vector are denoted by $\tilde{\boldsymbol{q}}^{(0)} = \left[\boldsymbol{q}^{l\,(0)}, \boldsymbol{q}^{u\,(0)}\right]$.

4.2.2 Step 2: Bounding linearization of interval nonlinear terms

This step aims at enclosing in a convex set $\mathcal{S}$ the nonlinear uncertain functions that exist in Problem (3.14). This will allow the formulation of a Linear Program. Problem (3.14) contains the nonlinear uncertain function $\tilde{f}_i(q_i)$ which is a function of one bounded variable. The bounds on these variables have been calculated in Step 1, such that for flow variables $q_i \in \left[q_i^l, q_i^u\right]$ and for head variables $h_j \in \left[h_j^l, h_j^u\right]$. The goal is to construct convex sets $\mathcal{S}$ that include all the points of the uncertain functions in the given range, i.e., $\tilde{f}_i(q_i) \in \mathcal{S}$, $\forall \tilde{q}_i \in \left[q_i^l, q_i^u\right]$, $i \in \mathcal{L}$.

In this work we design the convex sets $\mathcal{S}$ using bounding linearization [142]. This procedure can be used on any uncertain nonlinear function of one bounded variable and it encloses the function between two lines. For example, given the nonlinear uncertain function $\tilde{f}_i(q_i)$ for an interval $q_i \in \left[q_i^l, q_i^u\right]$, a convex region containing all points defined by this function can be designed as follows:

$$f^l_{L,i}(q_i) \leq \tilde{f}_i(q_i) \leq f^u_{L,i}(q_i),\ \forall q_i \in \left[q_i^l, q_i^u\right],\ i \in \mathcal{L} \tag{4.5}$$

where the lower line $f^l_{L,i}(q_i) \triangleq \lambda^l_i q_i + \beta^l_i$ and the upper line $f^u_{L,i}(q_i) \triangleq \lambda^u_i q_i + \beta^u_i$. The goal of the bounding linearization procedure is to define the line parameters to minimize the area of the resulting convex set $\mathcal{S}$. In the case of pipes, $\tilde{f}_i\left(q_i\right)$ is inclusion isotonic, the procedure needs to deal only with the two following ”boundary” functions:

$$\begin{aligned} f^l_i(q_i) &= r^l_i |q_i|^{\nu-1} q_i \\ f^u_i(q_i) &= r^u_i |q_i|^{\nu-1} q_i. \end{aligned} \tag{4.6}$$

First the procedure to define the lower line $f^l_{L,i}(q_i)$ is described. The slope λ^l_i is calculated using the lower two points of $\tilde{f}_i\left(q_i\right)$ at the boundaries q^l_i and q^u_i:

$$\lambda^l_i = \frac{\min\left(f^l_i\left(q^u_i\right), f^u_i\left(q^u_i\right)\right) - \min\left(f^l_i\left(q^l_i\right), f^u_i\left(q^l_i\right)\right)}{q^u_i - q^l_i}. \tag{4.7}$$

To minimize the area of the convex set but also to guarantee that the resulting convex set includes all the points defined by the nonlinear interval function $\tilde{f}_i(q_i)$, we have to make sure that (4.5) is satisfied. Since the slope of the line is defined by (4.7), the offset β^l_i is defined by solving analytically the following optimization problem, which uses (4.5) as constraints:

$$\beta^l_i = \left\{ \begin{array}{ll} \max & \beta_i \\ s.t. & \left\{ \begin{array}{l} f^l_{L,i}(q_i) \leqslant f^l_i(q_i) \\ f^l_{L,i}(q_i) \leqslant f^u_i(q_i) \\ q^l_i \leqslant q_i \leqslant q^u_i \end{array} \right. \end{array} \right. \tag{4.8}$$

The problem in (4.8) is linear in the cost function but nonlinear in the constraints, thus the Karush-Kuhn-Tucker optimality conditions should be satisfied. The procedure for finding the upper line $f^u_{L,i}(q_i)$ parameters is similar; the upper line slope λ^u_i is given by (4.9) and the offset β^u_i is given by (4.10):

$$\lambda^u_i = \frac{\max\left(f^l_i\left(q^u_i\right), f^u_i\left(q^u_i\right)\right) - \max\left(f^l_i\left(q^l_i\right), f^u_i\left(q^l_i\right)\right)}{q^u_i - q^l_i}. \tag{4.9}$$

$$\beta_i^u = \begin{cases} \min & \beta_i \\ \text{s.t.} & \begin{cases} f_{L,i}^u(q_i) \geqslant f_i^l(q_i) \\ f_{L,i}^u(q_i) \geqslant f_i^u(q_i) \\ q_i^l \leqslant q_i \leqslant q_i^u \end{cases} \end{cases} \tag{4.10}$$

The convex set defined by the lines in the range $q_i \in \left[q_i^l, q_i^u\right]$, will be referred to as the *approximated uncertainty area*.

We used the benchmark network "Net1" to illustrate the behaviour of successive bounding linearizations of the head-loss function $\tilde{f}_i(q_i)$ when using the IHISE algorithm on a WDN. We look into the head-loss function of pipe 6 of "Net1", given by $\tilde{f}_6(q_6)$, for different ranges of the flow q_6. In this function, the uncertain parameter is $r_6 \in [8.5\times10^{-3}, 9.4\times10^{-3}]$. As shown in Fig. 4.1, the water flow q_6 ranges from $q_6 \in [-1090, 1105]$ at the first iteration, to $q_6 \in [47.2, 60.5]$ at the sixth iteration. Notice that as the range becomes smaller, the approximation of the interval function $\tilde{f}_6(q_6)$ from the lines $f_{L,6}^l(q_6)$ and $f_{L,6}^u(q_6)$ becomes tighter.

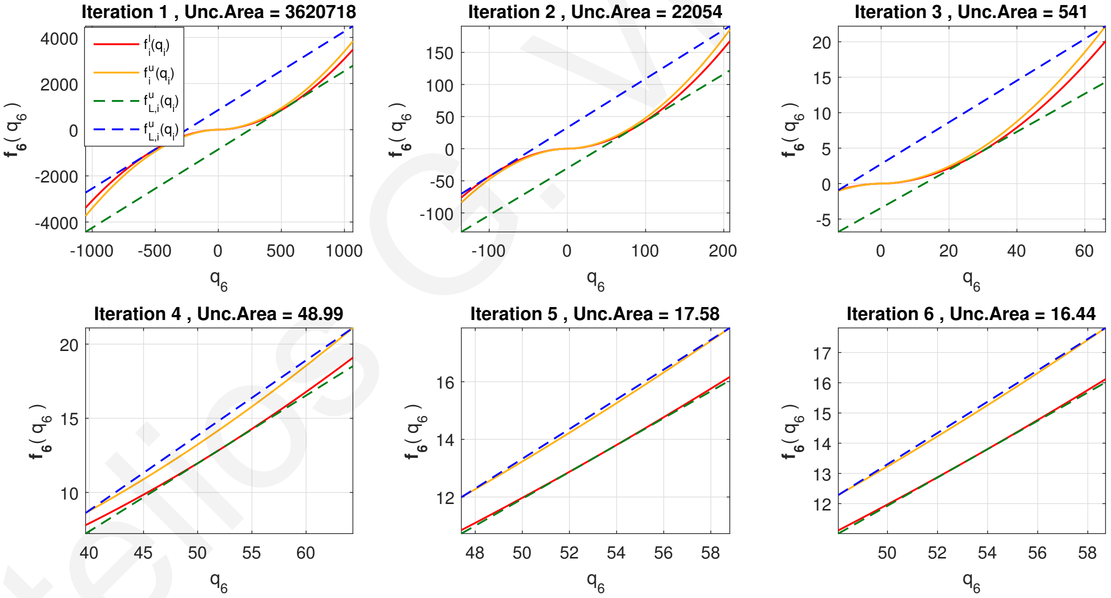


Figure 4.1. Bounding linearization of the uncertain head-loss function $\tilde{f}_6(q_6)$, for different ranges of the water flow q_6. At each iteration the uncertainty area becomes smaller, approximating the uncertainty of $\tilde{f}_6(q_6)$.

The uncertainty in the pressure-driven demand function $f_{ext,j}(h_j)$ is due to the uncertain head h_j. The uncertainty on the requested water demands $q_{ext,j}$ is represented by a positive interval for the requested demand of each node j:

$$\tilde{q}_{ext,j} = \left[q_{ext,j}^l, q_{ext,j}^u\right] \tag{4.11}$$

The uncertainty that should be considered in the case of leakage modeling, is on the leakage parameters, which can be represented by an interval $\tilde{\beta}_j = \left[\beta_j^l, \beta_j^u\right]$ and $\tilde{\gamma}_j = \left[\gamma_j^l, \gamma_j^u\right]$. This will result in an interval leakage function $\tilde{q}_{leak,j}(h_j)$.

Bounding linearization can also be applied on the pressure-driven functions $\tilde{q}_{ext,j}(h_j) \triangleq \tilde{q}_{ext,j} f_{ext,j}(h_j)$ and leakage functions $\tilde{q}_{leak,j}(h_j)$ as illustrated in Figure 4.2 where bounding linearization is applied on an uncertain pressure-driven demand function (center) and an uncertain leakage function (right) for a head range $h_j \in [4, 10](\mathrm{m})$. Notice that in the defined range, the bounding linearization procedure on these functions can benefit from the additional linear constraints, $\tilde{q}_{ext,j}(h_j) \triangleq \tilde{q}_{ext,j} f_{ext,j}(h_j) \geq 0$ and $\tilde{q}_{leak,j}(h_j) \geq 0$.

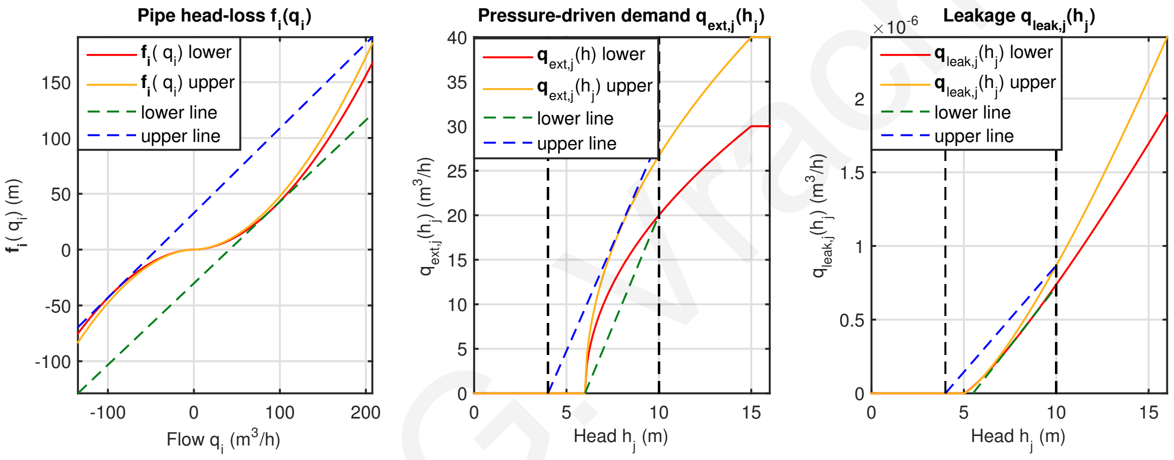


Figure 4.2. Illustration of bounding linearization of uncertain pipe head-loss function (left) in the flow range $q_i \in [-120, 205](m^3/h)$, of a pressure-driven demand function (center) and a leakage function (right) for a head range $h_j \in [4, 10](m)$.

4.2.3 Step 3: Formulation of Linear Program

The bounding linearization of Step 2, yields linear inequality constraints for the interval function $\tilde{f}_i(q_i)$. These inequalities can replace this functions in (3.14) with new variables ζ, thus transforming these equations into linear inequalities. Bound inequalities also arise on state variables q_i and h_j through Step 1. The Linear Program will then have the following constraints:

$$h^l_{ext,i} \leq \zeta + \sum_{j \in \mathcal{N}_u} \left(B_{ij}\, h_j \right) \leq h^u_{ext,i}, \qquad i \in \mathcal{L} \tag{4.12α′}$$

$$q^l_{ext,j} \leq \sum_{i \in \mathcal{L}} \left(B^\top_{ij}\, q_i \right) \leq q^u_{ext,j}, \qquad j \in \mathcal{N}_u \tag{4.12β′}$$

$$\lambda^l q_i + \beta^l \leq \zeta \leq \lambda^u q_i + \beta^u, \qquad i \in \mathcal{L} \tag{4.12γ′}$$

$$q^l_i \leq q_i \leq q^u_i \qquad i \in \mathcal{L} \tag{4.12δ′}$$

$$h^l_j \leq h_j \leq h^u_j \qquad j \in \mathcal{N}_u \tag{4.12ε′}$$

Note that the interval parameters in (3.14α′) are eliminated through the use of their upper and lower bounds to convert them into the inequalities (4.12α′) and (4.12β′). The LP decision variables vector is defined as $[\boldsymbol{x}^\top, \boldsymbol{\zeta}^\top]^\top \in \mathbb{R}^{(2n_l+3n_u)}$ where $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{h}^\top]^\top$ is the state vector and $\boldsymbol{\zeta}$ is the auxiliary variable vector. Using the constraints (4.12α′) - (4.12ε′), two LP problems can be formulated for obtaining lower (LPmin) and upper (LPmax) bounds on each state x_i:

LPmin:

$$\left\{ \begin{array}{ll} \min\limits_{[\boldsymbol{x},\boldsymbol{\zeta}]} & x_i \\ s.t. & (4.12α′) - (4.12ε′) \end{array} \right\}$$

LPmax:

$$\left\{ \begin{array}{ll} \max\limits_{[\boldsymbol{x},\boldsymbol{\zeta}]} & x_i \\ s.t. & (4.12α′) - (4.12ε′) \end{array} \right\}$$

4.2.4 Step 4: Solution of the linear interval system of equations

The objective of the optimization problem formulated in the previous section is to find the lower and upper bounds on the state vector $\boldsymbol{x}$ that satisfy the inequalities (4.12α′)-(4.12ε′). To achieve this, a total of $2(n_l + n_u)$ LPs must be solved where each problem will derive either the lower or upper bound of an individual state variable, indicated by x^*_i. This procedure is described in Algorithm 1.

4.2.5 Step 5: Iterative solution of the nonlinear interval system of equations

In Algorithm 1 the linearized version of the original problem in (3.14) is solved. This is an outer interval solution to the nonlinear problem, which guarantees to include the interval hull solution. To find the smallest possible interval $\tilde{\boldsymbol{x}} = \left[\boldsymbol{x}^l,\ \boldsymbol{x}^u\right]$ that satisfies (3.14), an iterative method is used. At each iteration m, the range $\tilde{\boldsymbol{x}}_{bnd}$ in which the optimization algorithm

Algorithm 1 Solution of LISE using LP

```
begin
for $i = 1$ to $n_l + n_u$ do
    Minimize $x_i$ by solving problem LPmin
    $x_i^l = x_i^*$
    Maximize $x_i$ by solving problem LPmax
    $x_i^u = x_i^*$
end for
$\tilde{\boldsymbol{x}} = \left[\boldsymbol{x}^l,\, \boldsymbol{x}^u\right]$
return $\bar{\mathbf{x}}$
```

searches for an optimal solution becomes smaller and is redefined as $\tilde{\boldsymbol{x}}_{bnd}^{(m+1)} = \tilde{\boldsymbol{x}}_{bnd}^{(m)} \cap \bar{\mathbf{x}}^{(m+1)}$, where $\tilde{\boldsymbol{x}}^{(m+1)}$ are the bounds calculated for the state vector $\boldsymbol{x}$ at iteration m. The iterations stop when the bounds on the state vector remain relatively unchanged, i.e., the change is smaller than a small number ϵ. The relative change in bounds is computed as follows:

$$e^{(m)} \triangleq \left|\left(\boldsymbol{x}^{u\,(m)} - \boldsymbol{x}^{l\,(m)}\right) - \left(\boldsymbol{x}^{u\,(m-1)} - \boldsymbol{x}^{l\,(m-1)}\right)\right|_1. \tag{4.13}$$

The overall procedure for the solution of the NISE is described in Algorithm 2. The resulting bounds are an OI solution of these equations that approximate closely the IH solution.

Algorithm 2 Iterative solution of NISE

```
begin
Define initial bounds $\bar{\mathbf{h}}^{(0)}$ using physical properties.
Calculate initial bounds $\bar{\mathbf{q}}^{(0)}$ using the procedure in Step 1.
$\bar{\mathbf{x}}_{bnd}^{(0)} = \left[\bar{\mathbf{q}}^{(0)^\top}\ \bar{\mathbf{h}}^{(0)^\top}\right]^\top$
$m = 0$
while $e^{(m)} > \epsilon$ do
    Bounding linearization of (3.14) for $\bar{x} \in \bar{\mathbf{x}}_{bnd}^{(m)}$
    Formulate problems LPmin and LPmax
    Find $\bar{\mathbf{x}}^{(m+1)}$ using Algorithm 1
    $\bar{\mathbf{x}}_{bnd}^{(m+1)} = \bar{\mathbf{x}}_{bnd}^{(m)} \cap \bar{\mathbf{x}}^{(m+1)}$
    $m = m + 1$;
end while
return $\bar{\mathbf{x}}^{(m)}$
```

4.3 Hydraulic interval-state observer

When dynamic states corresponding to tank levels exist in the system, the hydraulic model is described by the discrete-time state-space representation of (3.20). The algebraic equations of (3.20β′) are a nonlinear interval system of equations which must be solved in order to calculate $\boldsymbol{x}_2$. A specialized solver that is able to find the interval solution for such equations is the IHISE algorithm described in section 4.2, which we define as follows:

$$\left[\boldsymbol{x}_2^l(k), \boldsymbol{x}_2^u(k)\right] = f_s^d\left(\boldsymbol{x}_1(k), \boldsymbol{u}(k), \tilde{\boldsymbol{d}}(k); \tilde{\boldsymbol{\theta}},\right), \tag{4.14}$$

where $\left[\boldsymbol{x}_2^l(k), \boldsymbol{x}_2^u(k)\right]$ defines a vector of continuous intervals for the algebraic states, with $\boldsymbol{x}_2^l(k)$ being the lower bounds and $\boldsymbol{x}_2^u(k)$ being the upper bounds. The bounds on measurements of algebraic states, considering also measurement noise, are calculated as follows:

$$[\boldsymbol{y}_2^l(k), \boldsymbol{y}_2^u(k)] = \left[\boldsymbol{x}_2^l(k) - \boldsymbol{v}^u, \boldsymbol{x}_2^u(k) + \boldsymbol{v}^u\right] \tag{4.15}$$

The dynamic states corresponding to tank levels may be partially measured. Using the initial conditions (water level) of these states a Luenberger linear observer can be designed that gives an estimate of these states. The observer should use a non-linear solver to solve the algebraic equations; the observer structure is illustrated in Fig. 4.3. When using the interval-state estimator of (4.14) for the algebraic states, then an *interval observer* can be designed for tank states. The interval observer consists of two sub-observers, calculating the upper and lower estimate of the state, assuming that bounds on input uncertainty and sensor noise are available. In the design of the observer the disturbance and noise are integrated in the variable they affect, resulting in that variable being bounded. The observer error dynamics are given by $\boldsymbol{e}(k+1) = \left(A^d_{x_1} - L_d C_1\right)\boldsymbol{e}(k)$, where $\boldsymbol{e}(k) = \boldsymbol{y}(k) - \hat{\boldsymbol{y}}(k)$ is the error vector. The estimation error converges to zero when $\left(A^d_{x_1} - L_d C_1\right)$ has its eigenvalues inside the unit circle, thus L_d is chosen accordingly:

$$\begin{aligned}\hat{\boldsymbol{x}}_1^l(k+1) &= A^d_{x_1}\hat{\boldsymbol{x}}_1^l(k) + A^d_{x_2}\hat{\boldsymbol{x}}_2^l(k) + K^d_{x_1}\boldsymbol{u}(k) + G^d_{x_1}\boldsymbol{d}^l(k) + L_d(\boldsymbol{y}_1(k) - \hat{\boldsymbol{y}}_1^l(k)) \\ \hat{\boldsymbol{y}}_1^l(k) &= C_1\hat{\boldsymbol{x}}_1^l(k) + \boldsymbol{v}^l(k).\end{aligned} \tag{4.16}$$

$$\begin{aligned}\hat{\boldsymbol{x}}_1^u(k+1) &= A^d_{x_1}\hat{\boldsymbol{x}}_1^u(k) + A^d_{x_2}\hat{\boldsymbol{x}}_2^u(k) + K^d_{x_1}\boldsymbol{u}(k) + G^d_{x_1}\boldsymbol{d}^u(k) + L_d(\boldsymbol{y}_1(k) - \hat{\boldsymbol{y}}_1^u(k)) \\ \hat{\boldsymbol{y}}_1^u(k) &= C_1\hat{\boldsymbol{x}}_1^u(k) + \boldsymbol{v}^u(k).\end{aligned} \tag{4.17}$$

where $\boldsymbol{v}^l(k)$ and $\boldsymbol{v}^u(k)$ are the sensor noise vector lower and upper bound respectively, $\boldsymbol{d}^l(k)$ and $\boldsymbol{d}^u(k)$ are the disturbance lower and upper bounds respectively. The estimates $\hat{\boldsymbol{y}}_1^l(k)$ and $\hat{\boldsymbol{y}}_1^u(k)$ will converge to a lower and upper bound respectively for the sensor measurement

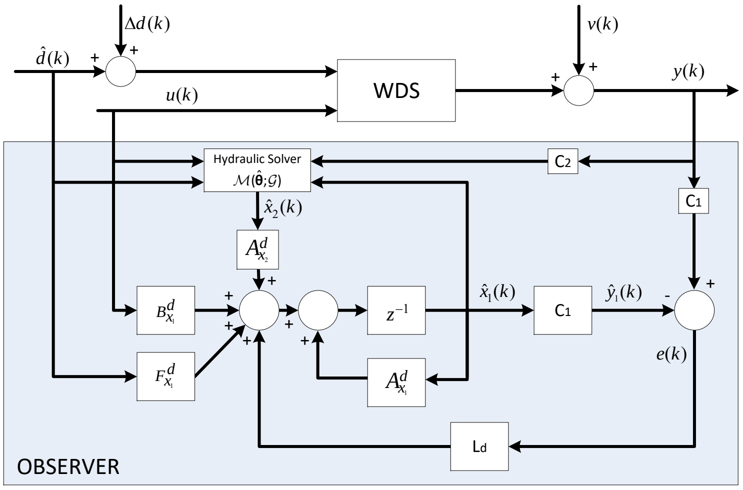


Figure 4.3. Discrete time observer for a WDS hydraulic state

vector $\boldsymbol{y}_1(k)$, as shown in [147], assuming that the system of interest is time-invariant and non-negative. This condition holds for the dynamic-state equations of this system.

4.4 Case Studies

In this section, the IHISE algorithm is applied on different WDS in order to demonstrate the calculated bounds and evaluate the performance of the algorithm. An demonstration of the concept is given in the “Illustrative example” subsection, where the bounds on different states are shown graphically and compared with bounds obtained by MCS. In subsection “Results from benchmark networks” a more extensive analysis of the algorithm is presented, as it is applied on different benchmark networks with varying characteristics. The performance of the algorithm is evaluated by defining appropriate performance metrics and by comparing the IHISE bounds with bounds obtained by MCS. In the simulations we assumed a demand-driven modeling approach with no leakages, which translates into $f_{ext,j}(h_j) = 1$ and $q_{leak,j}(h_j) = 0, \; j \in \mathcal{N}_u$ as in Problem (3.14). This modeling approach allows us to evaluate the performance of the algorithm using the established WDS simulation software EPANET [74].

4.4.1 Illustrative example

The benchmark network “Net1” shown in Fig. 4.4 provided by EPANET, is used to demonstrate the bounds on hydraulic states produced by the IHISE algorithm. The network parameters are shown in Table 4.1. Realistic water demand patterns, are assigned at each

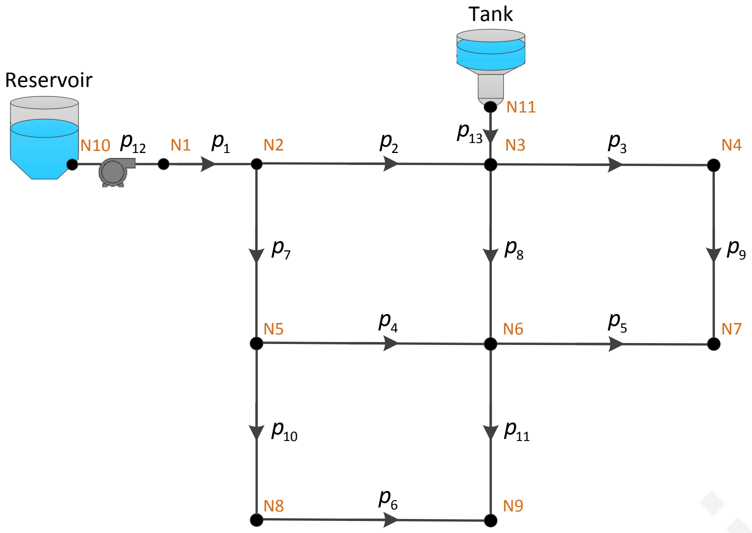


Figure 4.4. The benchmark network "Net1", on which the IHISE algorithm is demonstrated.

demand node.

The IHISE algorithm is used to generate bounds on water flows in pipes and hydraulic heads at nodes of the network. The measurement uncertainty is defined as ±5% on the given water demands at nodes, which is the typical error given by manufacturers of water flow meters. Modeling uncertainty is also considered and it is defined as ±5% on pipe Hazen-Williams coefficients. The simulation duration is 24 hours, with a discrete time step of one hour.

Additionally, the same bounds are generated using Monte-Carlo Simulations (MCS) of the network in EPANET. The demands are randomly varied at each simulation within a range of ±5% of the given water demands at nodes. The uncertainty on pipe Hazen-Williams coefficients is achieved by varying pipe lengths, as the Hazen-Williams coefficients are linearly proportional on this parameter. Uncertainty on pipe roughness coefficients and pipe diameter can also be considered, but the effect on Hazen-Williams coefficient will not be linear. The maximum and minimum value of each state is saved, defining the upper and lower bounds. The number of simulations is set to 30 000, at which point the rate of change of the MCS derived bounds was observed to converge to zero. Note that MCS provide an inner approximation of the bounds on each state and how close they are to the true bounds depends on the number of simulations. Given the possible variations of the same network for the given uncertainty, a sufficiently large number of simulations need to be performed in order for the MCS to converge to the true bounds. Nevertheless, especially in small networks, the MCS bounds can be useful to evaluate the IHISE bounds, which are an outer approximation of the

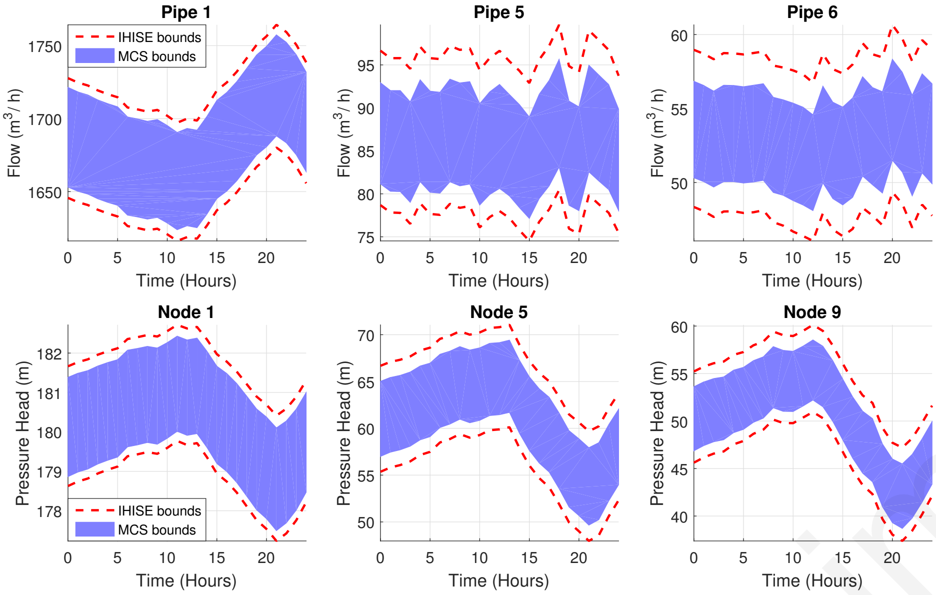


Figure 4.5. Comparison of selected pipes water flow bounds (above) and selected nodes hydraulic head bounds (below), generated by Monte-Carlo Simulations (blue solid area) and the IHISE algorithm (red dashed lines).

true bounds, by: 1) verifying the correctness of the IHISE bounds by checking if the MCS bounds are always a subset of the IHISE bounds and 2) evaluating the conservativeness of the IHISE bounds by measuring their distance from the MCS bounds.

Simulation results for selected states which reflect the results for all the states are given in Fig. 4.5. The IHISE bounds are compared with bounds generated using MCS for each state. The figure illustrates that the MCS bounds are a subset of the IHISE bounds, while they are also closely approximated. Note that, the true unknown bounds are enclosed between the IHISE and MCS upper and lower bounds.

4.4.2 Results from benchmark networks

To evaluate the ability of IHISE to compute state bounds, five benchmark networks with varying characteristics are used: "Anytown", which was used as the basis for the original "Battle of the Network Optimization Models" [148], "Net1", "Net2" and "Net3", which are example networks in EPANET [74], and "ky3" from the Kentucky Infrastructure Authority database of water distribution models [149]. The networks and their characteristics are listed in Table 4.1.

The networks were carefully selected to have multiple varying characteristics in terms of size, topology and types of elements they contain. Varying the network size, i.e., the number of nodes and links, demonstrates the scalability of the IHISE algorithm by considering the performance in networks with different number of *states*, i.e., heads at nodes and water flows at links. Varying the number of reservoirs and tanks, as well as the number of pumps in the network, reveals the ability of the algorithm to deal with these components. The topology of the networks is also considered, specifically the complexity that arises in calculating hydraulic states when the networks contain loops. This is quantified by calculating the *circuit rank* of the network indicated by γ, which is then normalized by the number of links n_l of each network. The resulting metric is defined as the *Loop Ratio*, given by $LR = \gamma/n_l,\ 0 \leq LR < 1$. A value of LR equal to zero means that there are no loops in the network, while LR approaches the value of one in the case of a fully connected graph.

Note that the circuit rank of an undirected graph is defined as the number of independent cycles, or the minimum number of edges that must be removed from the graph to break all its cycles making it into a tree. It is calculated as $r = m - n + c$, where m is the number of edges in the given graph, n is the number of vertices and c is the number of connected components. The circuit rank is also known as the cyclomatic number and is used to indicate the complexity of a program's source code [150].

Monte-Carlo and IHISE Simulations

For each network, a random demand scenario is assigned which produces a feasible solution in EPANET, i.e., there are no negative pressures. Similar to the "Illustrative example" section, the demand and modeling uncertainty is consider equal to ±5%. The simulation duration for all networks is 24 hours, with a discrete time step of one hour. The total number of simulation steps is defined as $N_s = 24$.

MCS were performed for all networks using EPANET. The varying parameters are the nodal demands and pipe parameters, in the range defined by the assumed uncertainty. At each simulation, the minimum and maximum value of each state for each time step is saved, thus defining the lower and upper bounds on the state. Additionally, a robust estimate of the state under the considered uncertainties is calculated by taking all the simulated scenarios and calculating the mean value of each state at each time step. This is indicated by $q^{\mu}_{MC,i}(k)$ for flow states and $h^{\mu}_{MC,j}(k)$ for head states and will be referred to as the *MCS state estimate*.

In order to perform an appropriate number of MCS and obtain quantifiable bounds for each network, a stopping criterion for the simulations is imposed, such that the change in all upper and lower flow-state and head-state bounds are less than $\Delta q(m^3/h)$ and $\Delta h(m)$ respectively, for at least $5\,000$ consecutive simulations. The flows and heads in each network may belong in different value ranges, so Δq and Δh are calculated as a percentage of the absolute mean value of the MCS state estimates. The absolute mean value of flow-states and head-states respectively is given by:

$$\begin{aligned} \mu_q &= \frac{1}{N_s} \sum_{k=1}^{N_s} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} |q^{\mu}_{MC,i}(k)| \right) \\ \mu_h &= \frac{1}{N_s} \sum_{k=1}^{N_s} \left(\frac{1}{n_n} \sum_{j=1}^{n_n} |h^{\mu}_{MC,j}(k)| \right) \end{aligned} \tag{4.18}$$

The defined accuracies are then calculated as $\Delta q = 1\%(\mu_q)$ and $\Delta h = 1\%(\mu_h)$. Using this approach, it is assumed that the bound accuracy is given by Δq and Δh. Note that while this approach provides a degree of confidence for the accuracy of MCS bounds, it is not guaranteed that the deviation of these bounds from the actual bounds cannot be larger.

Using the IHISE algorithm, bounds for the state of all networks are computed, using the assumed uncertainty. As a technical note, due to the fact that the IHISE algorithm was designed from scratch without the use of other hydraulic solvers, the networks had to satisfy specific conditions in order for the current version of the algorithm to work and be compared with the results from EPANET. The tank levels have to be measurable, so any tanks in the network model are replaced with variable head reservoirs. Control rules that open and close pumps depending on tank levels were removed from the model. Additionally, the head-loss formula should be set to Hazen-Williams. These limitations will be removed in future versions of the algorithm.

Evaluation of bounds mean value

An estimated value of the each state at each time step k is derived using the mean value of the IHISE bounds, which we will refer to as the *IHISE state estimate*. For flow-states this is calculated as $q^{\mu}_{IH,i}(k) = \left(q^{u}_{IH,i}(k) + q^{l}_{IH,i}(k)\right)/2$, and for head-states this is calculated as $h^{\mu}_{IH,j}(k) = \left(h^{u}_{IH,j}(k) + h^{l}_{IH,j}(k)\right)/2$. The IHISE state estimates are compared with the MCS state estimates, by calculating the Absolute Percentage Deviation (APD) of the IHISE state estimates to the MCS state estimates. Note that data from time steps where the MCS state estimate is close to zero are excluded from the evaluation, as they produced large percentages that are not representative of the results. The comparison shows that, for all networks, the Mean APD was less than 1%, while 99% of the time, the APD was less than 8%. This indicates that, when no statistical characterization of the uncertainties is available, the mean value of the IHISE algorithm bounds can be used as a robust estimate of the system state.

Evaluation of bounds

For the evaluation of the IHISE algorithm bounds, we compare the lower bounds $x^l_{IH} = [q^l_{IH}\ h^l_{IH}]$ and upper bounds $x^u_{IH} = [q^u_{IH}\ h^u_{IH}]$ of the IHISE algorithm with the lower bounds $x^l_{MC} = [q^l_{MC}\ h^l_{MC}]$ and upper bounds $x^u_{MC} = [q^u_{MC}\ h^u_{MC}]$ derived from MCS respectively. First, the validity of the IHISE bounds was checked, i.e., $x^l_{IH,i}(k) < x^l_{MC,i}(k)$ and $x^u_{IH,i}(k) > x^u_{MC,i}(k)$ for all states i and all time steps k. The test indicated that there were no bound violations for networks “Net1”, “Net2” and “Anytown”. For “Net3”, bound violations occur for two flow-states, in time steps when the MCS flow estimate is less than $10^{-4}(m^3/h)$ and the MCS bound width is less than $10^{-2}(m^3/h)$. For “ky3”, bound violations occur in 1.77% of the time, in specific states where the difference in width of the IHISE bounds and MCS bounds is less than $0.4(m^3/h)$ for flows and less than $0.005(m)$ for heads. The violation magnitude for any state and time step is less than 0.5% of the corresponding MCS state estimate. All the observed bound violations occur in cases where the IHISE bounds are very close to the MCS bounds. This can be explained by the fact that the IHISE algorithm uses a completely independent hydraulic solver than EPANET, thus differences in solutions may exist due to numerical errors or slight differences in modeling. Despite this fact, the differences in solutions made apparent by the violations are insignificant and the validity test of the IHISE can be considered successful if these are attributed to modeling uncertainty which has not been taken into account.

Next, the Absolute Deviation (AD) of the two sets of bounds is evaluated separately for flow-states and head-states, as they are measured in different units. The AD for flow-state lower and upper bound is defined as $e^u_{q,i}(k) = q^u_{IH,i}(k) - q^u_{MC,i}(k)$ and $e^l_{q,i}(k) = q^l_{MC,i}(k) - q^l_{IH,i}(k)$ respectively. Similarly, the AD for head-state lower and upper bounds is defined as $e^l_{h,j}(k)$ and $e^u_{h,j}(k)$ respectively. An illustration of lower and upper bound ADs is shown in Fig. 4.6. In Table 4.1 the mean AD for all states and time steps are shown for each network. The results indicate mean errors for upper and lower bounds that are close to the accuracy of MCS, ΔQ and Δh.

The area defined by the IHISE bounds is also an important evaluation metric. Since the duration of simulations is the same for all states, evaluating the area is equivalent to evaluating the width of the bounds. The width of the bounds for each time step is defined as the difference between the upper and lower bound for each time step. For IHISE flow-state bounds, for state i and time step k, the width is given by $w^q_{IH,i}(k) = q^u_{IH,i}(k) - q^l_{IH,i}(k)$. Similarly, the width of IHISE head-state bounds, for state j and time step k, is defined as $w^h_{IH,j}(k)$. The corresponding

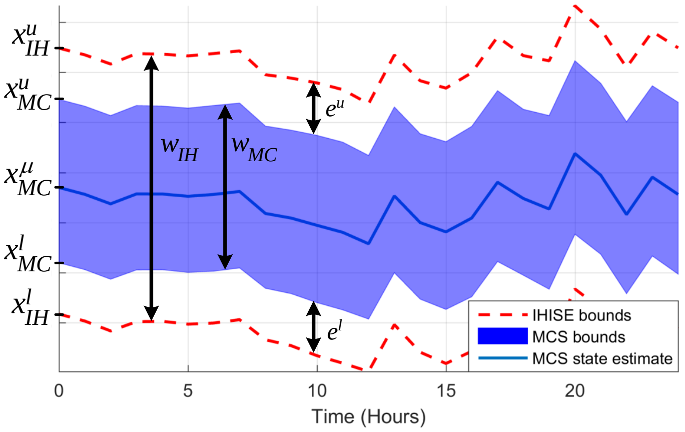


Figure 4.6. Illustration of bound evaluation parameters.

widths for MCS bounds are denoted by $w_{MC,i}^{q}(k)$ and $w_{MC,j}^{h}(k)$. An illustration of bound widths is given in Fig. 4.6 In Table 4.1 the mean bound widths for all states and time steps are shown for each network. The mean IHISE bound widths is indicated by w_{IH}^{q} for flow-states and w_{IH}^{h} for heads-states, while the MCS bounds are similarly indicated by w_{MC}^{q} and w_{MC}^{h}. As shown in Table 4.1, the difference between IHISE and MCS mean bound width for flow-states in different networks varies from $0.6(m^3/h)$ in "Net2" to $37.5(m^3/h)$ in "Net3". Similarly, the mean bound width for head-states varies from $0.04(m)$ in "Net2" to $2.05(m)$ in "Net1".

In order for the bounds width to give meaningful insight into the accuracy of the algorithm, they must be normalized relative to the absolute mean value of states for each network, i.e., μ_q for flow-states and μ_h for head-states. Using this normalization, the bound width can be viewed as a percentage of each state's uncertainty. The *Percent State Uncertainty (PSU)* is calculated using the bound width-to-mean ratio as follows:

$$\eta_{alg}^{s} \in \pm \frac{\left(w_{alg}^{s}/2\right)}{\mu_s} 100\%, \tag{4.19}$$

where $alg = \{IH, MC\}$ depending on the algorithm used, and $s \in \{q, h\}$ depending on the type of state. This will allow the comparison between the calculated state uncertainty and the uncertainty on the network inputs, i.e., the demand and parameter uncertainty. It is recalled that the uncertainty on demands and parameters is defined as a percentage of their estimated value, which was set at ±5% in these simulations.

The average PSU for each network, indicated by $\eta_{alg}^{s} : alg \in \{IH, MC\}, s \in \{q, h\}$, is given in Table 4.1. For flow states, the PSU is, for both methods, close to the ±5% input uncertainty which will be used as a reference point. Typically MCS have slightly less uncertainty and IHISE slightly more, with the exclusion of the looped network "Anytown" where both methods have more uncertainty, and the small network "Net1" where both methods have more. For head-states, the results are much different, as both methods produce much less state uncertainty than the reference point, except in the case of "Net1" where the uncertainty is near ±5%. The IHISE average PSU is at the worst case 2 times larger than the MCS average PSU. The worst cases present at the large network "ky3" but also in the highly looped network "Anytown".

For additional insight, the maximum PSU for each network is calculated. This is calculated using the MCS state estimate for each state i and time step k as follows: $\max(\eta^s_{alg}) = \max(w^s_{alg,i}(k)/s^\mu_i(k)) : alg = \{IH, MC\}, s = \{q, h\}$. Note that time steps when MCS state estimates have values close to zero, were excluded from the evaluation as they produced large percentages that are not representative of the results. As observed in Table 4.1, the IHISE maximum PSU is at worst 3 times larger than the maximum PSU obtained by MCS. The maximum values of PSU occur in only a few occasions, as it is illustrated in Fig. 4.7, where the distribution of the PSU for IHISE and MCS is plotted for network "ky3". From the figure it is observed that the bulk of IHISE PSU values are small and close to the MCS PSU values. Additionally, the number of occurrences seems to decline exponentially with the increase of PSU value, indicating that the PSU values calculated by the IHISE algorithm are bounded.

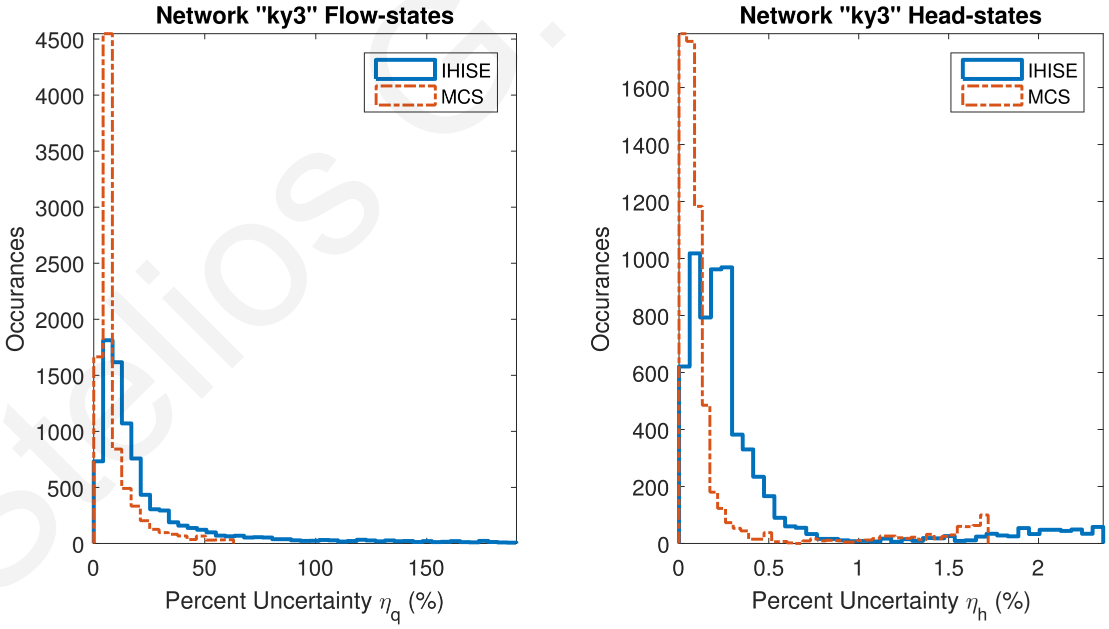


Figure 4.7. Distribution of Percent State Uncertainty (PSU) η for network "ky3".

The different operating scenarios of the networks resulting by the changing demands may also affect the bounds of the IHISE algorithm. To evaluate this factor, the average difference in PSU for all flow-states $\bar{\eta}^q_{IH}(k) - \bar{\eta}^q_{MC}(k)$ at each time step k is calculated. This is then compared to the average nodal demand in the network $\bar{q}_{ext}(k)$ at each time step. By performing correlation analysis, we obtain a correlation of 0.9483, 0.9995, 1.0000, 0.9273 and 0.9975 between these data, for the networks “Net1”, “Anytown”, “Net2”, “Net3” and “ky3” respectively. The correlation between these data is illustrated by plotting the average difference in PSU as a function of the average nodal demand for network “ky3”. The average difference in PSU follows the pattern of average nodal demands. This can be explained by the fact that the uncertainty on demands is proportional to the demand value, as it was assumed in the design of the simulations, and MCS bounds become less accurate when uncertainty in the network is larger, thus deviating more from the IHISE bounds. The correlation is illustrated in Fig. 4.8 where the average difference in PSU is plotted with the average nodal demand for network “ky3”. From the figure we observe that the average difference in PSU follows the pattern of average nodal demands. This can be explained by the fact that the uncertainty on demands is proportional to the demand value, as it was assumed in the design of the simulations, and MCS bounds become less accurate when uncertainty in the network is larger, thus deviating more from the IHISE bounds.

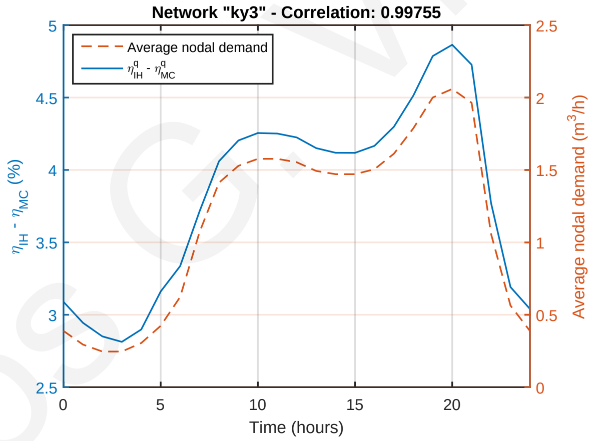


Figure 4.8. Correlation between average nodal demand in network “ky3” and the difference in percent flow-state uncertainty $\eta_{IH} - \eta_{MC}$.

Simulation times

The simulation *time* of either the IHISE algorithm or MCS for a single time step is also evaluated, along with the average *iterations* needed by the IHISE algorithm to solve a single time instance of the specific network and the number of MCS. The simulations were performed on a personal computer with Intel Core i5-2400 CPU at 3.10GHz. Simulation times of the IHISE algorithm are mainly depended on the size of the network, as observed in Table

4.1. An extrapolation of the simulation times for the IHISE algorithm compared to the simulation times of Monte-Carlo Simulations based on the five benchmark networks of this case study is shown in Fig. 4.9. The estimated simulation time of the IHISE algorithm is always less than the MCS time with the defined accuracy, while the time difference becomes larger for larger networks.

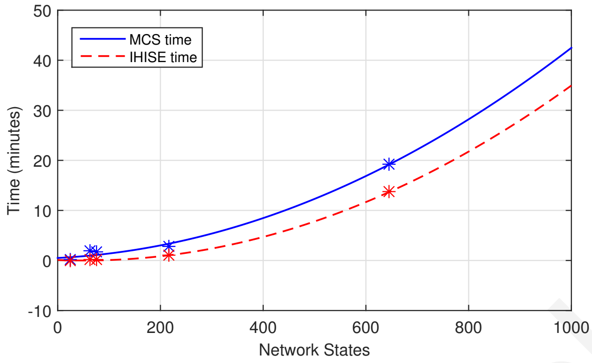


Figure 4.9. Extrapolation of simulation times for the IHISE algorithm and for MCS, as a function of the network size.

The simulation time also depends on the complexity of the network, as it is evident in Table 4.1 from the simulation time of the looped network “Anytown”. The IHISE algorithm needs more iterations to converge to a solution for this network compared to the ”Net2” which has similar number of states but is less looped. Similarly, more MCS are needed for the looped network “Anytown” than ”Net2” for them to converge to a defined bound accuracy.

4.5 Conclusions

In this work the problem of estimating bounds on WDS hydraulic states is addressed. A new methodology is proposed that generates interval state estimates. The proposed *Iterative Hydraulic Interval State Estimation* (IHISE) algorithm generates bounds on hydraulic states of the network, by taking into account the water demand uncertainty and modeling uncertainty in the form of uncertain pipe parameters. The uncertainties are modeled as intervals. The results show that the proposed methodology is able to generate tight bounds on hydraulic states and can be used in place of randomized methods such as Monte-Carlo Simulations (MCS).

The advantage of this methodology over MCS is that the calculated bounds guarantee the inclusion of the true system state, while the iterative nature of the algorithm makes these bounds as tight as possible. An extension of this work is to use the generated bounds for fault diagnosis methods that detect and localize leakages in the network.

The proposed methodology can be naturally used with model based fault-diagnosis and robust control methodologies, because many of these rely on the availability of bounding state estimates which are calculated by some knowledge of the system uncertainties. In the case of fault-diagnosis, the bounding state estimates are used to create thresholds, which when violated is an indication of a fault [151]. Additionally, the bounds on hydraulic states of the network can be used to generate bounds on water quality states, since the dynamics of hydraulic and quality states of a water network are interconnected.

Another application of the proposed methodology is for sensor placement with the objective to minimize state-estimation uncertainty. The IHISE algorithm, in its essence, propagates the uncertainty form water demands and model parameters into the interval-state estimates of each flow and head state in the network. In this respect, the bound widths of the derived interval-state estimates are an indication of network locations that are in need of monitoring by sensors in order to reduce state estimation uncertainty. An approach for sensor placement using the IHISE needs an available offline network model with demand pseudo-measurements to produce interval-state estimates. Then a pre-specified number of sensors can be placed sequentially, with the IHISE repeating after each sensor placement to redefine state uncertainty.

A limitation of this methodology is that it does not model elements whose head function is depended on pressure, such as pressure reduction valves. This is something that will be considered in future work. Other elements that are used in WDS and are not modeled in this work, are pressure control valves, flow control valves etc. Future work will model a variety of additional components to be used with this methodology, and an interval hydraulic state estimation toolkit will be released. Additionally, more extensive simulations on how this methodology deals with pressure-driven demands and pressure-dependent leakages will be provided.

Networks:	**Net1**	**Anytown**	**Net2**	**Net3**	**ky3**
States	24	63	76	216	646
Loop Ratio	0.23	0.49	0.13	0.19	0.26
Junctions	9	19	35	92	269
Reservoirs	1	3	0	2	3
Tanks	1	0	1	3	3
Pipes	12	40	40	117	366
Pumps	1	1	0	2	5
Flow-states:					
μ_q (m^3/h)	551.98	75.93	13.33	469.02	42.12
Δq (m^3/h)	5.52	0.76	0.13	4.69	0.42
e_q^u (m^3/h)	5.77	6.49	0.32	19.01	1.60
e_q^l (m^3/h)	6.82	6.54	0.32	20.04	1.62
w_{MC}^q (m^3/h)	32.55	10.85	0.96	42.64	2.91
w_{IH}^q (m^3/h)	42.75	23.54	1.56	80.15	6.08
η_{MC}^q (%)	±2.95	±7.15	±3.59	±4.55	±3.46
η_{IH}^q (%)	±3.87	±15.50	±5.86	±8.54	±7.22
max(η_{MC}^q) (%)	±19.33	±78.62	±23.12	±95.71	±62.53
max(η_{IH}^q) (%)	±37.86	±205.80	±40.52	±190.35	±190.07
Head-states:					
μ_h (m)	63.06	42.92	43.69	52.89	48.04
Δh (m)	0.52	0.37	0.41	0.50	0.49
e_h^u (m)	1.34	0.06	0.02	0.74	0.08
e_h^l (m)	1.35	0.07	0.02	0.61	0.09
w_{MC}^h (m)	6.10	0.11	0.03	1.49	0.29
w_{IH}^h (m)	8.16	0.24	0.07	2.76	0.46
η_{MC}^h(%)	±4.84	±0.13	±0.04	±1.41	±0.30
η_{IH}^h (%)	±6.47	±0.28	±0.08	±2.61	±0.48
max(η_{MC}^h) (%)	±12.99	±0.29	±0.10	±1.82	±1.71
max(η_{IH}^h) (%)	±17.44	±0.61	±0.21	±3.60	±2.35
Times:					
MCS Number	5849	28993	21805	13977	32695
MCS (min)	1.71	3.84	4.36	11.74	34.49
IHISE (min)	0.01	0.12	0.08	0.96	13.72
IHISE Iterations	7.44	13.48	9.00	14.56	16.40

Table 4.1. Results of the IHISE algorithm on benchmark networks.

Chapter 5

Leakage detection using interval-state estimation

In this section we describe a methodology for leakage detection in WDS by directly using hydraulic interval-state estimates derived by the IHISE algorithm as detection thresholds. The methodology is applied on a case study of a modified transport network in Cyprus, using real data, with the aim to determine the existence of unaccounted-for water. The results show that the existence of an artificial leakage into the data can be detected, due to state bound violation. The methodology is then evaluated using the LeakDB.

5.1 Methodology

During the operation of a WDS, measurements from flow and pressure sensors $\boldsymbol{y}(k) \in \mathbb{R}^{n_s}$ are also available as defined by (6.6). These measurements are related to the true state as follows:

$$\boldsymbol{y}(k) = C\left(\boldsymbol{x}(k) - z_0\right) + \boldsymbol{v}(k), \tag{5.1}$$

where $C \in \mathbb{R}^{n_s \times (n_n + n_l)}$ identifies the measured states, $\boldsymbol{x}(k)$ is the true and unknown state of the system, $z_0 = [\mathbf{0}^\top \boldsymbol{z}^\top]^\top$ contains the elevations needed to convert pressure measurement into head and $\boldsymbol{v}(k)$ is the measurement noise vector which is bounded by known bounds, i.e., $\boldsymbol{v}(k) \in \tilde{\boldsymbol{v}}(k)$.

The procedure of hydraulic interval-state estimation as described in Chapter 4, yields bounds on the system states, indicated by $\tilde{\boldsymbol{x}}(k) = [\tilde{\boldsymbol{q}}(k)^\top \; \tilde{\boldsymbol{h}}(k)^\top]^\top$, using only the inputs of the system. These inputs are the known hydraulic inputs $\boldsymbol{u}(k)$ and the uncertain uncontrolled input $\tilde{\boldsymbol{d}}(k)$ which includes the uncertain demands $\tilde{\boldsymbol{q}}_{ext}(k)$ and known head states $\boldsymbol{h}_{ext}$.

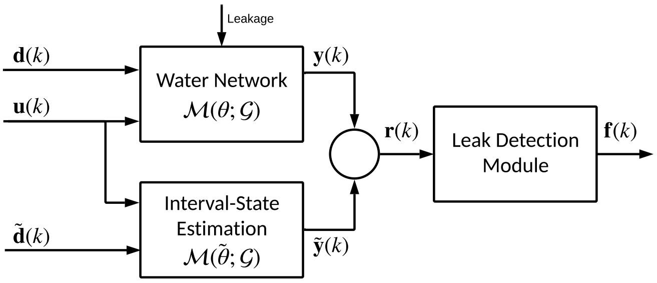


Figure 5.1. Illustration of the leakage detection procedure for WDS

The logic behind this leakage detection methodology is that each measurement from $\boldsymbol{y}(k)$ should be included in the corresponding state bounds derived by interval-state estimation. This is because interval-state estimates are derived using historical input data from the network in healthy operation, and moreover, the major sources of uncertainty have been considered. If any of the measurements is outside the calculated bounds, then these set of measurements should contain the effect of a fault in the system which has not been modeled during interval-state estimation. The aforementioned logic is applied on (5.1) in which the interval-state estimates replace the actual state. During normal operation the following should hold:

$$\boldsymbol{y}(k) \in \left\{C\left(\boldsymbol{x}\tilde{(k)} - z_0\right) + \tilde{\boldsymbol{v}}(k)\right\} \tag{5.2}$$

The condition in (5.2) may be violated for certain values of the measurements. To check for a violation we formulate the following residual vector using the lower and upper bounds of the interval quantities in (5.2):

$$r(k) = \begin{bmatrix} \boldsymbol{y}(k) - \left[C\left(\boldsymbol{x}^u - z_0\right) + \boldsymbol{v}^u(k)\right] \\ \left[C\left(\boldsymbol{x}^l - z_0\right) + \boldsymbol{v}^l(k)\right] - \boldsymbol{y}(k) \end{bmatrix} \in \mathbb{R}^{2n_s}. \tag{5.3}$$

The residuals are then compared with the threshold $\boldsymbol{\gamma} = 0$ to express the normal operation condition, given by $\boldsymbol{r}(k) < \boldsymbol{\gamma}$. The value of γ can be increased to reduce false alarms in cases when there have been un-modeled uncertainties in the state estimation process. A *fault signature* vector $\boldsymbol{f}(k) \in \{0,1\}^{2n_s}$ is then defined to show violation of condition (5.2) at each time step:

$$f_i(\mathrm{k}) = \begin{cases} 0, & r_i(k) < \gamma \\ 1, & r_i(k) > \gamma \end{cases} , i \in \{1 \cdots 2n_s\}. \tag{5.4}$$

A fault is detected if there exists an $i \in \{1 \cdots 2n_s\}$ for which $f_i(k) = 1$. The fault detection procedure using interval-state estimation is illustrated in Fig. 5.1.

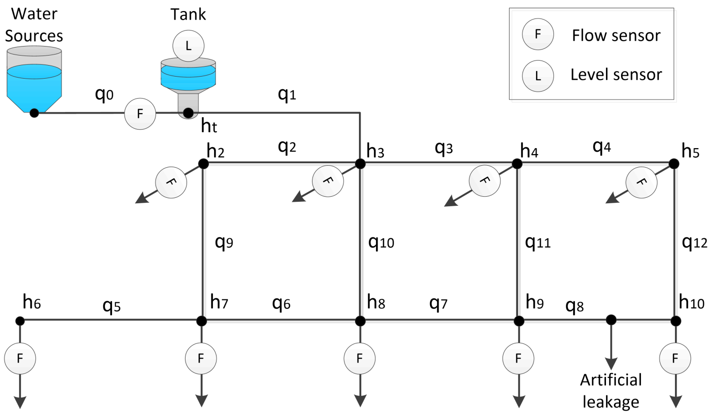


Figure 5.2. Illustrative diagram of the modified Limassol water transport sub-network of this case study.

5.2 Case study: Limassol, Cyprus

This study uses data from a real water transport sub-network in Limassol, which is the second largest city in Cyprus. A modified version of the transport network is used, of which an illustrative diagram is shown in Figure 5.2. The modified network contains three loops and it comprises of 9 demand nodes, one water tank and 12 links which represent pipes. Flow sensors (F) are installed at demand nodes, which represent aggregated real measurements at entrances to DMAs, and a water level sensor (L) is installed in the tank. Sensor measurements arrive at fixed five-minute intervals. The tank's water input originates from four water sources, of which three are water dams and one is a desalination unit. The water inflow q_0 coming from these sources is measured with a flow sensor. The water outflow q_1 of the tank is not directly measured.

5.2.1 Real-time implementation

The implementation of this case study in real-time is based on a platform for real-time monitoring of WDN against hydraulic and quality events. A model of the transport network was created as an EPANET input file. Using the platform, one can select the dates with available sensor data and request a state estimation. The available measurements from demand nodes and the level of the tank are then retrieved and a data validation process takes place which replaces missing data.

Sensor measurements have an uncertainty which is defined by the installed sensor's specifications. The measurements given by the flow sensors are within ±2% of the actual flow at those locations. Modeling uncertainty is also present in the form of pipe parameter uncer-

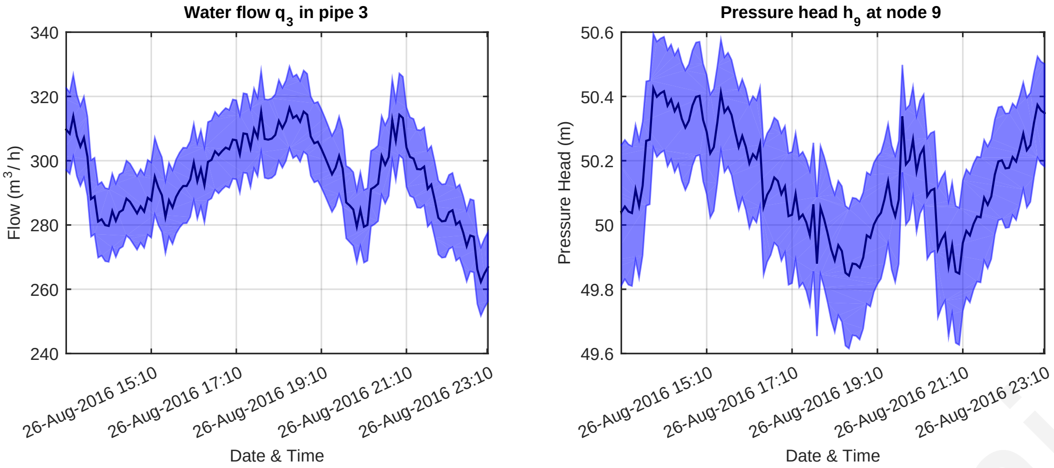


Figure 5.3. State estimate (black line) and bounds on this estimate using the IHISE algorithm (blue area) for the water flow in pipe 3 (left) and the head at node 9 (right).

tainty. For this case study we assumed a total uncertainty of ±2% on the Hazen-Williams coefficient. The value of this uncertainty may vary, as it is calculated using expert-elicited bounds on the modeled pipe parameters. It is assumed that the network graph is known, thus structural uncertainty is neglected. This is a valid assumption in transport networks where the structure is known, as it is the network in this case study.

Using the IHISE algorithm, bounds on water flows and pressures in the network are generated using the flow measurements at demand nodes and the tank level measurements, by taking into account measurement and modeling uncertainty. The algorithm needs approximately 4 seconds to calculate bounds for each hydraulic step, on a personal computer with Intel Core i5-2400 CPU at 3.10GHz. The bounds converge after 8 iterations. The size of bounds does not increase over time because it depends only on the measurements of that specific time step. The effect of accumulating uncertainty due to the dynamic calculation of tank levels does not affect the size of the bounds in this case study, because the tank level is measurable. For illustration purposes, flow and pressure estimates using a real-time EPANET-based state estimator are also generated. The state estimates for a selected pipe and node, accompanied by its corresponding uncertainty bounds generated by the IHISE algorithm, are shown in Figure 5.3.

5.2.2 Determining the existence of unaccounted-for water

A common practice in water utilities is to use mass balance to determine if there is unaccounted-for water exiting the network. In this case study, since there is no sensor mea-

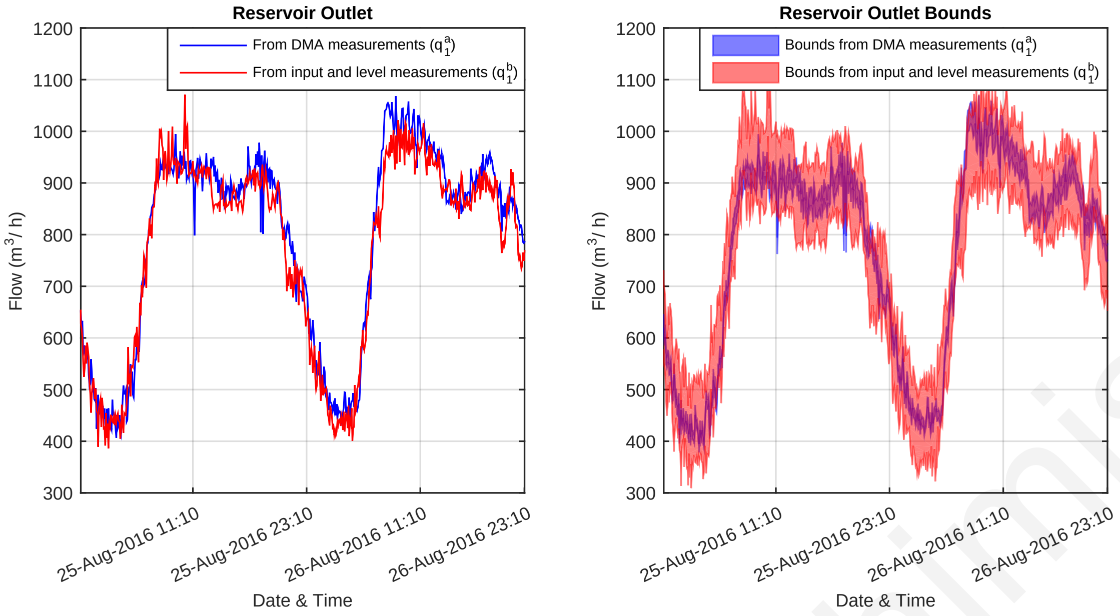


Figure 5.4. Left: Comparison of two estimates of the tank outflow, $q_1^a(k)$ and $q_1^b(k)$, Right: Comparison of the uncertainty bounds generated by the IHISE algorithm for the same estimates.

suring the tank outflow q_1, mass balance can be checked by generating an estimate of q_1 using two different sets of data: the first is by calculating the sum of all the measured outflow (demands), indicated here by $q_1^a(k)$; the second is the calculation of the tank outflow using the measured tank inflow $q_0(k)$ and tank water level measurement $h_t(k)$, indicated here by $q_1^b(k)$, as follows:

$$\begin{aligned} q_1^b(k) &= q_0(k) - (\alpha_t/\Delta t)\ \Delta h_t(k) \\ \Delta h_t(k) &= h_t(k) - h_t(k-1), \end{aligned} \tag{5.5}$$

where α_t is the base area of the tank and $\Delta t = 5$ minutes is the measurement time step.

Using data from the case study network, the two tank outflow estimates were calculated for a period of two days, from '24-Aug-2016 23:10' to '26-Aug-2016 23:10'. The two estimates are compared in Figure 5.4 (left). It can be observed that the two estimates have a non-zero difference at almost all time steps. This can be due to noisy data, and thus it cannot be determined with certainty if there is unaccounted-for water. A way to deal with this is to calculate the average difference e_d between these data for the given period of time, i.e., $e_d = mean\left\{(q_1^a(k) - q_1^b(k)), \forall\ k \in \{1...k_s\}\right\}$, where k_s is the total number of measurements from each sensor. This calculation gives a constant unaccounted-for flow $e_d = 18.82\ (m^3/h)$, which may be due to background leakages or it may be due to non-uniformly distributed measurement errors. By checking the water utility leakage reports of the examined period, there was no recorded leakage for the sub-network of this study.

Using the IHISE algorithm and the given model and measurement uncertainties, bounds on these same estimates can be calculated: the bounds on tank outflow by simulating the network using the network outflow and tank level measurements are indicated by $\boldsymbol{q}_1^a(k)$, and bounds on tank outflow using the tank inflow and tank level measurements are indicated by $\boldsymbol{q}_1^b(k)$. The comparison of these two sets of bounds is shown in Figure 5.4 (right). The two sets of bounds overlap, indicating that the variance can be explained by measurement and modeling uncertainty. There are some specific time steps that the bounds do not overlap, which may be due to noisy data that can be eliminated using a suitable data validation strategy. It can also be observed that bounds generated by the tank level and tank inflow measurements are wider. This is because these bounds are calculated using the dynamic equation (5.5) which uses three uncertain measurements for the calculation of the bounds.

5.2.3 Determining the existence of an artificial leakage

In this section the potential of the IHISE algorithm to be used for event detection in water distribution systems is demonstrated. An artificial leakage is added to the network model, at the location indicated in Figure 5.2. The leakage has a magnitude of $20\ (m^3/h)$ and its time profile is described by an abrupt constant outflow starting at ‘26-Aug-2016 00:10’.

In order to determine the location of the leak, additional measurements should be available. Assuming the existence of pressure sensors in the network, a comparison of the measured pressure with the estimated pressure could indicate the presence of a leak. However, in this case, the measurements are affected not only by the sensor uncertainty (as when calculating mass balance), but also by the network modeling uncertainty which may greatly affect the pressure estimates. Using the IHISE algorithm, the effect of both measurement and modeling uncertainties is considered in calculating the bounding estimates and the existence of a leak can be determined with greater certainty.

We assume the existence of a pressure sensor at node 9 of the network as shown in Figure 5.2. The pressure sensor reading is compared with the IHISE bounding estimates, as shown in Figure 5.5 (left). The error between the pressure sensor reading and the estimated bounds, which is defined as the distance of the reading from the bounds when the reading is outside the bounds, is also calculated and shown in Figure 5.5. It is observed that there is a pressure sensor reading error after the leakage occurs. The error presents only during the night hours, when the pressure is higher due to the low demand and thus a pressure drop due to a leakage

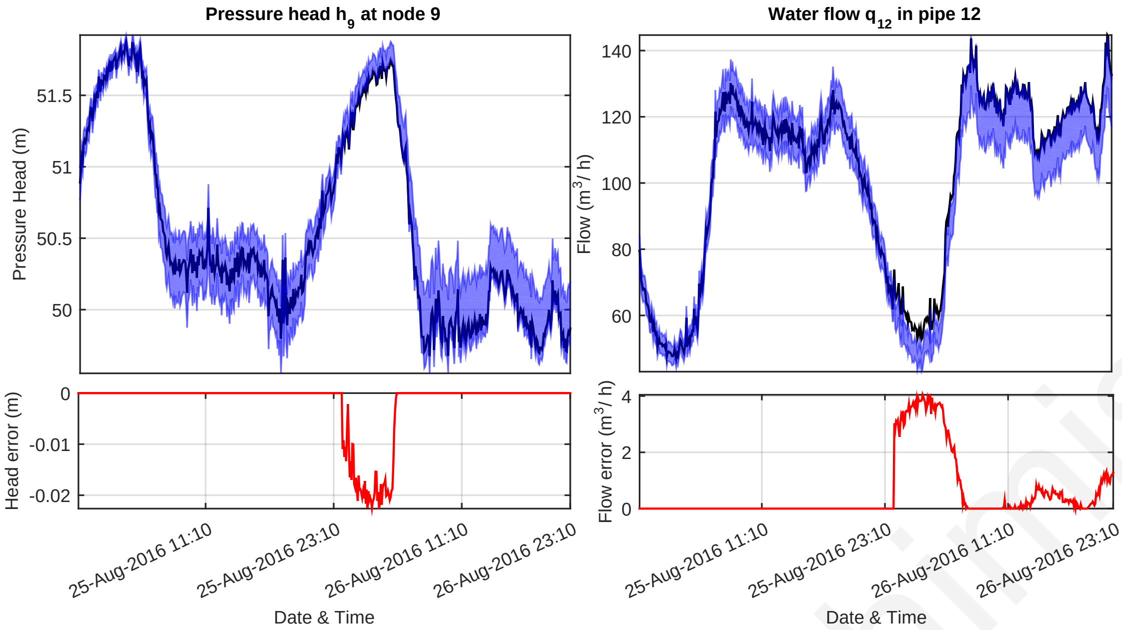


Figure 5.5. The effect of a leakage occurring in the network at time '26-Aug-2016 00:10' on a pressure (left) and a flow (right) state, compared to the estimated uncertainty bounds for the same states calculated by the IHISE algorithm. Below each graph, the corresponding error of the state compared to the calculated bounds is presented.

is more apparent. Similarly, if we assume the existence of a flow sensor on pipe 12, the same effect can be observed when the flow reading is compared with the IHISE bounds, as shown in Figure 5.5 (right). There is a flow sensor error during the night hours, while the error persists in smaller magnitude for the rest of the day. These observations indicate the existence of the leakage despite the measurement and modeling uncertainties in the network.

5.3 Methodology evaluation using LeakDB

The benchmark network "Net1", an example network in EPANET [74] shown in Fig. 4.4, is used to demonstrate the leakage detection methodology based on interval state estimation. EPANET is a public domain, water distribution system modeling software package developed by the United States Environmental Protection Agency's Water Supply and Water Resources Division. The network model uses realistic demand patterns for a period of 24 hours. We consider a known bounds on uncertain demands $\tilde{\boldsymbol{d}}(k)$ at each time step. Additionally, pipe parameter uncertainty is considered on the estimated pipe parameters which are also bounded by a known bound $\tilde{\boldsymbol{\theta}}^u$.

In total, 810 different scenarios are used from LeakDB in which the parameters and

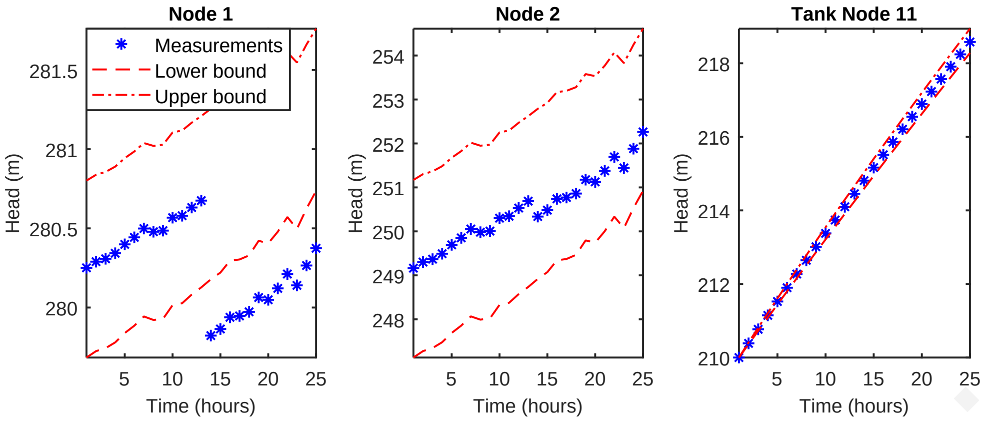


Figure 5.6. Bounded state estimates for three nodes of the WDS and the measurements from a leakage scenario, where a leakage occurs on the 14^{th} hour at node 1.

demands are varied randomly inside their uncertainty bounds. The demand uncertainty upper bounds $\Delta\boldsymbol{d}^u(k)$ and the parameter uncertainty upper bounds $\Delta\boldsymbol{\theta}^u$ were varied between $\pm 2\%$, $\pm 5\%$ and $\pm 10\%$ of the estimated demands $\hat{\boldsymbol{d}}(k)$ and estimated parameters $\hat{\boldsymbol{\theta}}$ respectively. Using the defined uncertainty bounds, interval state estimation is performed as described in Chapter 4. An example of the calculated state bounds is given in Fig. 5.6.

In each scenario, a leakage is induced on a node, which is random in location and magnitude. The leakages have a different profile, by varying leak hole diameter between $0cm$ (no leak) to $4.5cm$. The leak magnitude $q_{leak}(k)$ (m^3/h) depends on the circular hole area and the pressure at that location as follows:

$$q_{leak}(k) = cAp^a(k)\sqrt{\frac{2}{\rho}}, \tag{5.6}$$

where A is the hole area (m^2), c is the discharge coefficient (unitless), $p(k)$ (m) is the pressure at the node which is calculated using the head value, a is the discharge exponent, and ρ is the density of the fluid (in this case, water). The simulation scenarios where generated using the Water Network Tool for Resilience (WNTR), which is an open source Python package designed to help water utilities investigate resilience of water distribution systems to hazards [131].

Measurements arrive every hour from pressure sensors installed at the nodes and a level sensor installed in the tank. Measurements have random noise with an upper bound of $\pm 2\%$ of the measurement value. These are used for detecting leaks in the network. The level sensor measurement is also used in the dynamic tank state observer. Different sensor placement scenarios are considered, ranging from sensors covering 100% of the nodes (10 sensors) to covering 20% of the nodes (2 sensors). Sensor placement is performed using a greedy algorithm which prioritizes the sensor locations in which the most faults have been detected in the simulated scenarios.

The proposed detection algorithm using interval state estimation was evaluated by recording the cases in which: 1) A fault existed and was detected (True Positive TP), 2) A fault existed but was not detected (False Negative FN), 3) A fault did not exist but a fault was detected (False Positive FP), 4) A fault did not exist and was not detected (True Negative TN). Moreover, the F1-score is calculated, a popular metric in datasets characterized by class imbalances [11]. The percentages of each metric in relation to the number of pressure sensors in the network are presented in Table 5.1.

The first observation from applying the detection methodology is that there are no false alarms when a fault is not present (0% False Positives and 100% True Negatives). The explanation behind this result is the assumption that the upper bound on uncertainties is accurately known. As a result, the interval state estimation methodology produces state bounds that are guaranteed to include the true state. In this respect, the proposed methodology favors the avoidance of false alarms. The trade-off is that the conservative bounds produced by interval state estimation result in a lower percentage of True Positive detections and in accordance, there is a higher percentage of False Negatives. A less conservative selection of the upper bound on uncertainties would increase TP percentages, but would also introduce FP detections.

To get more insight into the results, TP percentages are plotted in Fig. 5.7 as a function of the leak hole diameter, for each one of the three uncertainty scenarios. The leak hole diameter is selected as the varying parameter because the leakage flow varies as a function of the node pressure, as in equation (5.6). Indicative leakage magnitudes are calculated by taking the average leakage flow in each simulation with the same leak hole area. This calculation reveals a flow of $2.6(m^3/h)$ for a hole size of $5(mm)$, $41(m^3/h)$ for a hole size of $2(cm)$ and $158(m^3/h)$ for a hole size of $4(cm)$. As it is intuitively expected, the TP percentage increases when the leak size is increased and when the uncertainty is lower. The results presented in Fig. 5.7 give an indication of the required accuracy in system parameters and inputs, in order to perform leakage detection.

Table 5.1. Statistics from applying the proposed fault detection algorithm to a total of 810 leakage scenarios of the LeakDB

#Sensors	TP %	FN %	FP %	TN %	F1 %
10	66.25	33.75	0	100	79.69
8	66.25	33.75	0	100	79.69
6	57.20	42.80	0	100	72.74
4	55.14	44.86	0	100	71.08
2	53.91	46.09	0	100	70.05

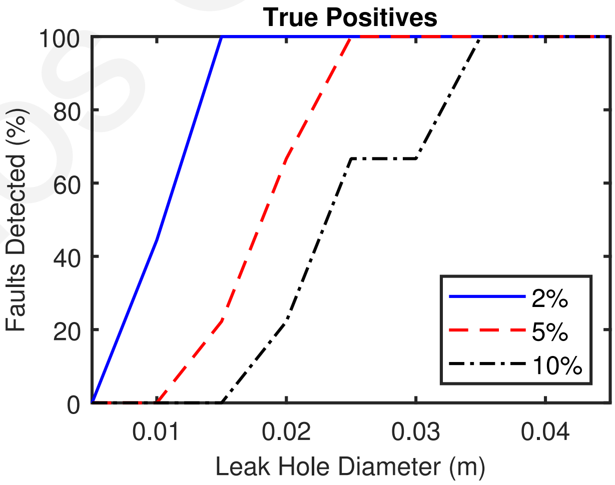


Figure 5.7. Detection percentages as a function of leak diameter, for each of the three cases of upper bounds on uncertainty (10 sensors).

Chapter 6

Leakage diagnosis using model invalidation

Model-based methodologies can assist in addressing the challenging problem of leakage detection and localization in water distribution systems. However, this is not trivial due to inherent non-linearities and parametric uncertainties. Most importantly, due to the small number of available sensor measurements compared to the number of system states, the inverse problem for estimating leakages is highly under-determined. In this work we propose the utilization of *a priori* available information about the system to formulate a hydraulic model of the system in its non-linear form, in which uncertainties are modeled by intervals defined by a lower and upper bound. A novel optimization-based methodology then utilizes pressure and flow measurements to perform leakage detection through model-invalidation. A modification of the optimization algorithm is activated in the case of a detection to refine possible leak locations and retain only the ones that can be explained by the interval model and available measurements from multiple time-steps. The proposed methodology is demonstrated on a benchmark network and evaluated using a leakage diagnosis benchmark dataset.

6.1 Introduction

Recent works on fault diagnosis have proposed the concept of set-membership model invalidation, in which data are used to create a set-valued non-faulty model of the system, and check its validity against future data using an online optimization formulation [152]. This formulation keeps implicit constraints to represent sets and the problem of fault-detection is reduced to the feasibility of a convex optimization problem [153], which can be checked using state-of-the-art solvers [154]. This method guarantees that there will be no false alarms, as long as the model of the non-faulty system remains valid. Moreover, there is no need to compute a decision threshold, while the flexibility in which constraints can be added into the formulation facilitates the utilization of additional system information to increase fault detectability.

In this work we utilize demand and model parameter bounds to formulate a "healthy" interval-model of the system in its nonlinear form, which also incorporates current measurements and estimated demands of node clusters derived from a demand calibration process. A novel optimization-based methodology for leakage detection through model invalidation is then proposed which detects inconsistencies between the healthy interval-model and sensor measurements. We show that the leakage detection problem in WDS can be reduced to iteratively examining the feasibility of a Linear Program (LP). In case of a leakage detection, a novel localization algorithm is activated to refine the possible leak locations and retain the ones that satisfy the constraints imposed by the interval-model and the available measurements from multiple time-steps.

The remainder of this chapter is organized as follows: In Section 6.2 the problem of formulating an interval-hydraulic-model using *a priori* available information is described. Then in Section 6.3 we introduce the methodology for leakage detection using interval-model invalidation. A novel leakage localization procedure is then described in Section 6.4 and the Localization Priority Index for evaluating this procedure is proposed in Section 6.5. The proposed methodologies are demonstrated using two illustrative examples in Section 7.4, and moreover they are extensively tested using a leakage benchmark dataset in the same Section. Finally, conclusions are drawn in Section 6.7.

Notation: Matrices are denoted with capital letters, vectors with lower bold letters, and scalars by italic letters. Sets and graphs are denoted by calligraphic letters. Uncertain parameters are represented by a continuous interval of values defined by a lower and upper bound. Intervals are accompanied by a tilde and defined as follows: $\tilde{\boldsymbol{v}} = [\boldsymbol{v}^l, \boldsymbol{v}^u]$ is a closed interval vector, where $\boldsymbol{v}^l$ is the lower bound vector and $\boldsymbol{v}^u$ is the upper bound vector, such that: $\tilde{\boldsymbol{v}} = \{\boldsymbol{v} \in \mathbb{R}^n : v_i^l \leq v_i \leq v_i^u, \forall\, i = \{1, .., n\}\}$, and n is the size of the vector.

6.2 Problem formulation

The topology of a WDN is modeled by a directed graph denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Let $\mathcal{N} = \{1, \cdots, n_n\}$ be the set of all nodes, where $|\mathcal{N}| = n_n$ is the total number of nodes. These represent junctions of pipes, consumer water demand locations, reservoirs and tanks. Moreover, let $\mathcal{N}_t \subset \mathcal{N}$ represent the subset of reservoir and tank nodes. The hydraulic state associated with nodes is the *hydraulic head*, indicated by h_j, $j \in \mathcal{N}$. The hydraulic head consists of a component analogous to the pressure p_j at node j, and of the node elevation z_j in respect to a geodesic reference. Each node j is also associated with a water consumer demand at the node location, denoted by $q_{\text{ext},j}$. Water demands drive the dynamics of a WDN and are typically an unknown input to the system. Leakages in this work are modeled at network nodes, such that $q_{\text{leak},j}$ represents a leakage occurring at node j. Let $\mathcal{L} = \{1, \cdots, n_l\}$ be the set of links, where $|\mathcal{L}| = n_l$ is the total number of links. These represent network pipes, water pumps and pipe valves. The hydraulic quantity associated with a link $i \in \mathcal{L}$ is the *water flow*, indicated by q_i. The hydraulic equations which describe the behavior of the system are the conservation of energy equations and the conservation of mass equations.

Energy in WDS is associated with the head at nodes and when water flows through a network link i which connects two nodes, a flow dependent, typically nonlinear head function $f_i(q_i)$ describes the change in head. The *energy equations* for all the network links, can be written as follows:

$$\mathbf{f}(\boldsymbol{q}) + B\boldsymbol{h} = \mathbf{0}, \tag{6.1}$$

where $B \in \mathbb{R}^{n_l \times n_n}$ is the incidence flow matrix, indicating the connectivity of nodes with links such that element $B_{ij} = +1$ if the conventional direction of link i enters node j; element $B_{ij} = -1$ if the conventional direction of link i leaves from node j; otherwise $B_{ij} = 0$. Moreover, the nonlinear mapping $\mathbf{f}(\boldsymbol{q}) : \mathbb{R}^{n_l} \mapsto \mathbb{R}^{n_l}$ represents the head-loss at links. Each function $f_i(q_i)$ represents the head-loss at link i, which is a measure of the energy dissipated due to friction of water flowing through the link. In the case of simple pipes, head-loss depends on the water flow through the pipe but also on pipe parameters, such as pipe length, diameter and pipe roughness coefficient. In this work, the Hazen-Williams (H-W) formula [127] is used to calculate head-loss in pipes, in which the effect of pipe parameters is aggregated in the H-W resistance coefficient r_i of each pipe. The head-loss across pipe $i \in \mathcal{L}_p$ is then calculated using the H-W formula as follows:

$$f_i(q_i) = r_i \left|q_i\right|^{(\nu-1)} q_i\,, \tag{6.2}$$

where ν is a constant exponent associated with the H-W formula and q_i is the water flow in pipe i. The head-loss function $f_i(q_i)$ can be adjusted to represent other system components modeled at links, such as pumps and valves [1].

The conservation of mass law for each node $j \in \mathcal{N}$ dictates that the sum of branch water flows from pipes incident to a node j must be equal to the node's external water demand plus any leakages associated with that node. The *conservation of mass equations*, considering all the nodes of the network, can be written using the incidence flow matrix as follows:

$$B^\top \boldsymbol{q} = \boldsymbol{q}_{\text{ext}} + \boldsymbol{q}_{\text{leak}}(\boldsymbol{h}). \tag{6.3}$$

The underlying modeling assumptions in (6.3) are:

1. The hydraulic heads at reservoir and tank nodes are measured and known with uncertainty which is within known bounds, such that:

$$\tilde{h}_j(k) \in \left[h_j^l(k), h_j^u(k)\right], j \in \mathcal{N}_t, \tag{6.4}$$

where $h_j^l(k)$ and $h_j^u(k)$ are known lower and upper bounds respectively for the head at node j.

2. The network is operating in pressure-sufficient conditions, such that the demand at each node is independent of the node pressure.

3. Leakage magnitude is pressure dependent. In this work, the pressure dependent leakage function is given by:

$$q_{\text{leak},j}(h_j) = c_j(h_j - z_j)^\alpha \tag{6.5}$$

where $c_j \geq 0$ is the leak emitter coefficient, $z_j \geq 0$ is the elevation at node j and $\alpha > 0$ is the leak emitter exponent.

The complete hydraulic-state of a WDS, comprised of the water flows in pipes and hydraulic heads at nodes, is indicated by $\boldsymbol{x} = [\boldsymbol{q}^\top\ \boldsymbol{h}^\top]^\top \in \mathbb{R}^{n_l+n_n}$. The state can be estimated using the hydraulic equations of (6.1) and (6.3) and available measurements from the system. In the case of a typical District Metered Area (DMA) of a WDS, flow $\hat{q}_j(k), j \in \mathcal{L}_{in} \subset \mathcal{L}$ measurements are available at the inlets at each discrete measurement time-step k, where $|\mathcal{L}_{in}| = n_i$ is the number of inlets. Additionally, a number of pressure sensors are installed inside the DMA, giving pressure measurements indicated by $\hat{p}_i(k), i \in \mathcal{N}_s \subset \mathcal{N}$, where $|\mathcal{N}_s| = n_s$ is the number of sensors. Sensor measurements include noise, which is assumed bounded, such that the true value of measured state relates to the measurements as follows:

$$C_q\boldsymbol{q}(k) = \hat{\boldsymbol{q}}(k) + \tilde{\boldsymbol{v}}_q(k) \tag{6.6α′}$$

$$C_h\boldsymbol{h}(k) = \hat{\boldsymbol{p}}(k) + \boldsymbol{z} + \tilde{\boldsymbol{v}}_p(k) \tag{6.6β′}$$

where $C_q \in \{0,1\}^{n_i \times n_l}$ and $C_h \in \{0,1\}^{n_s \times n_l}$ are matrices used to identify measured flow and head states respectively, and $\tilde{\boldsymbol{v}}_q$, $\tilde{\boldsymbol{v}}_p$ are the noise bounds for flow and pressure devices respectively.

WDS have typically a small number of sensors installed, i.e. $(n_s + n_i) \ll n_n$, which makes the system of equations defined by (6.1) and (6.3) under-determined. According to the analysis in [23], at least n_n sensors in an observable configuration are needed to guarantee the topological observability of the system. A possible observable configuration according to [23], is when water demands are measured at all nodes and at least one additional head measurement is available. For this reason demand estimates, called *pseudo-measurements*, are used to complement missing measurements and make the problem over-determined by utilizing in total more than n_n measurements [42]. Demand estimates can be derived using historical information such as population densities, building areas and the consumption patterns of typical consumers [136]. These estimates are then further improved by combining them with available low-resolution measurements, such as quarterly billing data. They can also be a function of the measured DMA inflow, as shown in [50]. Since it is impossible to obtain exact estimates, an alternative depiction of demand pseudo-measurements is in the form of error bounds [40], given for each discrete time-step k as follows:

$$\tilde{\boldsymbol{q}}_{\text{ext}}(k) = \left[\boldsymbol{q}^l_{\text{ext}}(k),\ \boldsymbol{q}^u_{\text{ext}}(k)\right]. \tag{6.7}$$

Demand calibration is the use of available historical sensor measurements, hydraulic equations and optimization methodologies to approximate the demand magnitude and behavior at each node as close as possible. Due to the small number of measurements, nodal demands with similar user characteristics are grouped (i.e., aggregated) to make the problem of demand calibration over-determined [46]. The procedure results in a demand estimate of n_g node groups, where $n_g \leq n_s$, which is less uncertain related to demands pseudo-measurements at nodes. The group estimate is written as:

$$M\boldsymbol{q}_{\text{ext}}(k) = \boldsymbol{g}(k), \tag{6.8}$$

where $M \in \{0,1\}^{n_g \times n_n}$ identifies demand groups, $\boldsymbol{g} \in \mathbb{R}^{n_g}$ is the uncertain group demand estimate. Notice that overlapping groups are allowed. The uncertainty on group estimates is considered by defining the bounds on these estimates, such that $\tilde{\boldsymbol{g}} \in \left[\boldsymbol{g}^l, \boldsymbol{g}^u\right]$. Moreover, considering a fault-free situation when $q_{\text{leak},j} = 0,\ \forall j \in \mathcal{N}$, (6.3) can be used to re-write (6.8) in terms of the system flow-states, as follows:

$$MB^\top \boldsymbol{q}(k) = \tilde{\boldsymbol{g}}(k) \tag{6.9}$$

Uncertainty in model parameters is modeled as an uncertain head-loss $\tilde{\mathbf{f}}(\boldsymbol{q})$, where the tilde indicates that this function contains uncertain parameters. Specifically for pipes, uncertainty is included in the uncertain H-W coefficients r_i. The pipe head-loss function for a pipe i, given by (6.2), when this contains uncertainty becomes:

$$\tilde{f_i}\,(q_i) = \tilde{r}_i\,|q_i|^{\nu-1} q_i, \quad i \in \mathcal{L}_p. \tag{6.10}$$

Model parameter bounds are assumed known and constant, because these parameters vary slowly over time. Note that this approach of modeling uncertainties can be applied to the head-loss function of any element of the system modeled by a link, as explained in [1].

6.3 Leakage detection by interval-model invalidation

The bounded uncertainties on demand and parameter estimates for every time-step k are derived from historical data of the system in healthy operation. During healthy operation, it is assumed that $q_{\text{leak},j} = 0,\ \forall j \in \mathcal{N}$ and measurements are included in the feasible solution set when solving the hydraulic equations considering bounded uncertainties and sensor noise. The fault-free interval-model for each discrete time-step k, is given by:

$$\tilde{\mathbf{f}}(\boldsymbol{q}(k)) + B\boldsymbol{h}(k) = 0 \tag{6.11α′}$$

$$B^\top \boldsymbol{q}(k) = \tilde{\boldsymbol{q}}_{\text{ext}}(k) \tag{6.11β′}$$

$$MB^\top \boldsymbol{q}(k) = \tilde{\boldsymbol{g}}(k) \tag{6.11γ′}$$

$$C_q \boldsymbol{q}(k) = \hat{\boldsymbol{q}}(k) + \tilde{\boldsymbol{v}}_q(k) \tag{6.11δ′}$$

$$C_h \boldsymbol{h}(k) = \hat{\boldsymbol{p}}(k) + \boldsymbol{z} + \tilde{\boldsymbol{v}}_p(k) \tag{6.11ε′}$$

The interval hydraulic model of (6.11) offers a framework for the design of an event detection methodology using model-invalidation. Unlike many set-membership methods that compute explicit set representations for the system states, a model-invalidation method uses an online optimization formulation that *keeps implicit constraints to represent sets*; it has been shown that model invalidation problems can be reduced to the feasibility of a linear optimization problem, which can be checked efficiently using state-of-the-art solvers [152].

In previous work, the authors have proposed a way to deal with hydraulic models which contain uncertainties in the form of intervals, by converting the nonlinear model into a set of linear inequalities and then using an iterative optimization procedure to calculate tight bounds on the system states [1]. However, the methodology could not deal with inconsistencies between the model and sensor measurements occurring during the presence of a fault;

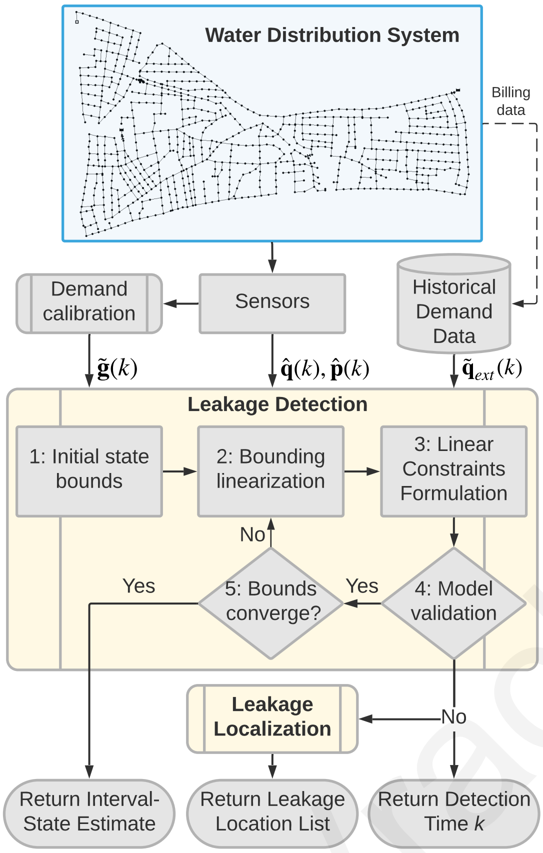

Water Distribution System
Billing data
Demand calibration
Sensors
Historical Demand Data
$\tilde{\mathbf{g}}(k)$
$\hat{\mathbf{q}}(k), \hat{\mathbf{p}}(k)$
$\tilde{\mathbf{q}}_{ext}(k)$
Leakage Detection
1: Initial state bounds
2: Bounding linearization
3: Linear Constraints Formulation
No
Yes
5: Bounds converge?
Yes
4: Model validation
Leakage Localization
No
Return Interval-State Estimate
Return Leakage Location List
Return Detection Time k

Figure 6.1. Illustration of the proposed leakage detection procedure.

consequently, when applying the aforementioned methodology for leakage detection in [12], the state bounds are calculated using offline demand pseudo-measurements during healthy operation and then compared to online sensor measurements to detect the presence of a fault.

In this work, we build on the previous work of [1, 12] to design a novel optimization procedure which includes both offline and online measurements and incorporates additional constraints imposed by demand calibration. The proposed model-based methodology for leakage detection through model invalidation exhibits a much higher detection rate than the one described in [12], as shown in the results of Section 6.6.2. Moreover, during healthy operation, the optimization procedure yields tighter bounds on state variables.

The improved performance can be explained by the fact that using model-invalidation the linear constraints formulated during the leakage detection procedure are part of the state-set representation, along with the bounds on states; in comparison, the state-set in [12] is represented only by the bounds on states.

The proposed leakage detection methodology is formulated in a series of five steps which are illustrated in Fig. 6.1 and explained below.

6.3.1 Step 1 : Initial bounds on state variables

The iterative leakage detection algorithm requires initial bounds on the state $\boldsymbol{x} = [\boldsymbol{q}^\top \, \boldsymbol{h}^\top]^\top$ as a starting point. This iterative procedure is applied to a single time-step. For notational convenience, the use of the time-step notation k is omitted. The initial bounds on the unknown head vector are chosen by considering physical properties of the network [1] and denoted by

$$\tilde{\boldsymbol{h}}^{(m)} = \left[\boldsymbol{h}^{l\,(m)},\ \boldsymbol{h}^{u\,(m)}\right], \tag{6.12}$$

where m indicates the iteration counter which for the initial bounds $m = 0$. Note that reservoir and tank node heads are represented here by their known bounds $\tilde{h}_j(k), j \in \mathcal{N}_t$. The initial bounds on the flow state vector, denoted by

$$\tilde{\boldsymbol{q}}^{(m)} = \left[\boldsymbol{q}^{l\,(m)}, \boldsymbol{q}^{u\,(m)}\right], \tag{6.13}$$

are calculated using $\tilde{\boldsymbol{h}}^{(m)}$ in equation (6.11α′) and solved with respect to flow states using interval arithmetic [135]. The initial bounds are intentionally conservative in order to ensure the inclusion of the true system state.

6.3.2 Step 2: Bounding Linearization

Bounding linearization is a procedure which encloses the non-linear uncertain functions $\tilde{f}_i\,(q_i)$ in a convex set, as described in [1]. The nonlinear uncertain functions contained in $\tilde{\mathbf{f}}(\boldsymbol{q}(k))$ of (6.11α′) are eliminated through the use of the auxiliary variables ζ_i at each diagonal position i. The convex set for each auxiliary variable ζ_i is defined using the bounds on the corresponding flows $\tilde{\boldsymbol{q}}^{(m)}$, thus it changes at each iteration m. The resulting linear inequality derived from bounding linearization, without denoting the iteration number m, is given by:

$$\lambda_i^l q_i + \beta_i^l \leq \zeta_i \leq \lambda_i^u q_i + \beta_i^u \tag{6.14}$$

where λ_i^l, β_i^l, λ_i^u and β_i^u are line parameters defined during bounding linearization. This procedure introduces uncertainty into the problem, as the intervals in (6.13) and the inequalities in (6.14) define a convex superset of the set defined by the function $\tilde{f}_i\,(q_i)$.

6.3.3 Step 3: Formulation of a Relaxed Problem

The bounding linearization procedure in section 6.3.2 enables us to rewrite problem (6.11) as a set of linear inequalities and eliminates the interval parameters through the use of their lower and upper bounds:

$$\boldsymbol{\zeta} + B\boldsymbol{h} = 0 \tag{6.15α'}$$

$$\Lambda^l \boldsymbol{q} + \boldsymbol{b}^l \leq \boldsymbol{\zeta} \leq \Lambda^u \boldsymbol{q} + \boldsymbol{b}^u \tag{6.15β'}$$

$$\boldsymbol{q}^l_{\text{ext}} \leq B^\top \boldsymbol{q} \leq \boldsymbol{q}^u_{\text{ext}} \tag{6.15γ'}$$

$$\boldsymbol{g}^l \leq MB^\top \boldsymbol{q} \leq \boldsymbol{g}^u \tag{6.15δ'}$$

$$\boldsymbol{v}^l_q \leq C_q \boldsymbol{q} - \hat{\boldsymbol{q}} \leq \boldsymbol{v}^u_q \tag{6.15ε'}$$

$$\boldsymbol{v}^l_p \leq C_h \boldsymbol{h} - \hat{\boldsymbol{p}} - \boldsymbol{z} \leq \boldsymbol{v}^u_p \tag{6.15$\sigma\tau'$}$$

$$\boldsymbol{q}^l \leq \boldsymbol{q} \leq \boldsymbol{q}^u \tag{6.15ζ'}$$

$$\boldsymbol{h}^l \leq \boldsymbol{h} \leq \boldsymbol{h}^u \tag{6.15η'}$$

The linear constraints in (6.15) can be used to formulate two Linear Programs (LP) for obtaining lower (LPmin) and upper (LPmax) bounds on each state variable x_i, as follows:

LPmin:

$$\left\{ \begin{array}{cc} \min\limits_{\boldsymbol{x}} & x_i \\ s.t. & (6.15) \end{array} \right\}$$

LPmax:

$$\left\{ \begin{array}{cc} \max\limits_{\boldsymbol{x}} & x_i \\ s.t. & (6.15) \end{array} \right\}$$

Problems **LPmin** and **LPmax** are a relaxed version of the original problem in (6.11), due to the bounding linearization of non-linearities. However, it is ensured that the feasible set which satisfies the original problem is a subset of the feasible set satisfying the relaxed problem.

6.3.4 Step 4 : Model Validation

The feasible set for the states $\boldsymbol{x}$ which satisfies the relaxed problem is found by solving a total of $2(n_l + n_n)$ LPs, with each problem calculating either the lower (**LPmin**) or upper (**LPmax**) bound of an individual state variable x_i. Assuming no faults in the system and valid uncertainty bounds, the constraints of (6.15) define a feasible set which contains the true system state and problems **LPmin** and **LPmax** can be solved for all state variables x_i. However, if a fault exists in the system, its effect will be present in the measurements and a feasible set may not exist, unless the fault effect is "concealed" by the combined demand, parameter and bounding linearization uncertainty. The decision that a feasible set does not exist is taken when any of the $2(n_l+n_n)$ LPs solved using a state-of-the-art solver [154] gives an infeasible solution.

6.3.5 Step 5 : Iterative Model Validation

To find the smallest possible interval $\tilde{\boldsymbol{x}} = \left[\boldsymbol{x}^{l},\ \boldsymbol{x}^{u}\right]$ that satisfies (6.11), an iterative method is used which reduces the bounding linearization uncertainty. At each iteration m, the tightest possible state vector bounds $\tilde{\boldsymbol{x}}^{(m)}$ given current constraints are calculated. At the next iteration $m+1$, these bounds are used in the bounding linearization procedure to calculate new constraints on the variable $\boldsymbol{\zeta}$. The new constraints are then in turn used to calculate new state vector bounds $\tilde{\boldsymbol{x}}^{(m+1)}$. As long as the linear program of Step 4 is able to be solved, the model is validated and iterations continue. The iterations stop when the bounds on the state vector remain relatively unchanged, in which case no fault is detected given time-step k measurements. The algorithm is considered to have converged when the L^1-norm of the change in bounds at iteration m, denoted by $e^{(m)}$ and defined in (6.16), is smaller than a small number ϵ.

$$e^{(m)} \triangleq \left|\left(\boldsymbol{x}^{u\,(m)} - \boldsymbol{x}^{l\,(m)}\right) - \left(\boldsymbol{x}^{u\,(m-1)} - \boldsymbol{x}^{l\,(m-1)}\right)\right|_1 . \tag{6.16}$$

The complete iterative algorithm for model-validation for each time-step k containing Steps 1–5 is outlined in *Algorithm 1*.

6.4 Leakage localization using an interval-model

When a leakage is detected using *Algorithm 1*, the interval-model is reformulated to consider the leakage localization problem. The interval-model of the system for each time-step k, considering the effect of leakages, is given by:

$$\tilde{\mathbf{f}}(\boldsymbol{q}(k)) + B\boldsymbol{h}(k) = 0 \tag{6.17α′}$$

$$B^{\top}\boldsymbol{q}(k) = \tilde{\boldsymbol{q}}_{\text{ext}}(k) + \boldsymbol{q}_{\text{leak}}(\boldsymbol{h}) \tag{6.17β′}$$

$$MB^{\top}\boldsymbol{q}(k) = \tilde{\boldsymbol{g}}(k) + M\boldsymbol{q}_{\text{leak}}(\boldsymbol{h}) \tag{6.17γ′}$$

$$C_q\boldsymbol{q}(k) = \hat{\boldsymbol{q}}(k) + \tilde{\boldsymbol{v}}_q(k) \tag{6.17δ′}$$

$$C_h\boldsymbol{h}(k) = \hat{\boldsymbol{p}}(k) + \boldsymbol{z} + \tilde{\boldsymbol{v}}_p(k) \tag{6.17ε′}$$

The leakage localization problem reduces to the determination of a feasible set for the augmented vector of state variables $\boldsymbol{\chi} = [\boldsymbol{x}^{\top}\boldsymbol{c}^{\top}]^{\top}$ given the interval model in (6.17) and assuming that only one leakage is present in the system. The new state variables $\boldsymbol{c}$ are the unknown leak emitter coefficients in the nonlinear functions $\boldsymbol{q}_{\text{leak}}(\boldsymbol{h})$ given in (6.5), and the feasible set for these variables indicates possible leakage locations. The leakage localization procedure is divided into the five steps described below.

Algorithm 1 Leakage detection algorithm

```
Input: time-step $k$, measurements $\hat{\boldsymbol{q}}(k)$, $\hat{\boldsymbol{p}}(k)$, demand bounds $\tilde{\boldsymbol{q}}_{\text{ext}}(k)$, $\tilde{\boldsymbol{g}}(k)$, parameter bounds $\tilde{r}$
Output: $\tilde{\boldsymbol{x}}(k)$, Detection

begin
Calculate initial state bounds $\tilde{\boldsymbol{x}}^{(0)}$
Iteration: $m = 0$
while $e^{(m)} > \epsilon$ do
    Bounding linearization using $\tilde{\boldsymbol{x}}^{(m)}$
    Formulate problems LPmin and LPmax
    for $i = 1$ to $n_l + n_n$ do
        Solve LPmin and LPmax to find $x_i^l$ and $x_i^u$
        if infeasible then
            Detection = True
            Go to Algorithm 2 (Leakage Localization)
        else
            $\tilde{x}_i^{(m)} = \left[x_i^l,\, x_i^u\right]$
        end if
    end for
    $m = m + 1$
end while
$\tilde{\boldsymbol{x}}(k) = \tilde{\boldsymbol{x}}^{(m)}$
return
```

6.4.1 Step 1 : Initial bounds on states and leakage functions

Initial bounds on the states $\boldsymbol{x}$ are calculated for iteration $m = 0$ as described in section 6.3.1. The initial bounds on the states $\boldsymbol{c}$ are calculated implicitly, by defining bounds on the leakage magnitudes $q_{\text{leak},j} = \left\|q_{\text{leak},j}(h_j)\right\| \; \forall j \in \mathcal{N}$. We assume that the existence of a leak is possible at any node, thus all nodes have a non-zero upper bound for the leakage flow, equal to $q^u_{\text{leak},j} = q^{max}_{\text{leak}} \, \forall j \in \mathcal{N}$. Similarly, the lower bound of all leakages is zero, such that leakage magnitude bounds become $\tilde{\boldsymbol{q}}_{\text{leak}} = \left[0, \boldsymbol{q}^{max}_{\text{leak}}\right]$. The maximum leakage magnitude can be roughly determined by methodologies that monitor and detect abnormalities of the system inflow [71]. Given leakage and state $\boldsymbol{x}$ bounds, bounds on the emitter coefficients $\boldsymbol{c}$ in (6.5) can be calculated using interval arithmetic, as follows:

$$\tilde{c}_j^{(m)} = \frac{\tilde{q}_{\text{leak},j}^{(m)}}{\left(\tilde{h}_j^{(m)} - z_j\right)^{\alpha}}, \; j \in \mathcal{N} \tag{6.18}$$

This work assumes pressure sufficient conditions in the network, thus $h_j > z_j$. The emitter exponent is assumed known and equal to $\alpha = 0.5$.

6.4.2 Step 2: Bounding linearization

The bounding linearization procedure is generalized to enclose in a convex set not only the uncertain head-loss functions $\tilde{f}_i\,(q_i)$ as described in Section 6.3.2, but also the uncertain nonlinear leakage functions $q_{\text{leak},j}(h_j)$. Each leakage function $q_{\text{leak},j}(h_j)$ is approximated through the use of the auxiliary scalar variable $q_{\text{leak},j}$ which corresponds to the leakage magnitude at node j. The additional linear inequality for each node j derived from bounding linearization is calculated using emitter coefficient bounds $\tilde{c}_j$ and head bounds $\tilde{h}_j$ and given by:

$$\lambda_{q,j}^{l} h_j + \beta_{q,j}^{l} \leq q_{leak_j} \leq \lambda_{q,j}^{u} h_j + \beta_{q,j}^{u}, \; j \in \mathcal{N} \tag{6.19}$$

where $\lambda_{q,j}^{l}$, $\beta_{q,j}^{l}$, $\lambda_{q,j}^{u}$ and $\beta_{q,j}^{u}$ are line parameters defined during bounding linearization.

6.4.3 Step 3: Formulation of the Relaxed Problem and calculation of leakage bounds

The elimination of nonlinearities through bounding linearization and the use of bounds enables the reformulation of Problem (6.17) as a set of linear inequalities, similar to Section 6.3.3. The decision variables will now include the auxiliary leakage variables $\boldsymbol{q}_{\text{leak}}$, such that the complete decision variables vector will be given by

$$\boldsymbol{z} = [\boldsymbol{x}^{\top} \; \boldsymbol{q}_{\text{leak}}^{\top}]^{\top}. \tag{6.20}$$

Here, however, the assumption of a single leakage existing in the system must be exploited, by considering that only one element $q_{\text{leak},j}$ $j \in \mathcal{N}$ is non-zero, which should satisfy the constraint $q^l_{\text{leak},j} \leq q_{\text{leak},j} \leq q^u_{\text{leak},j}$. The set of linear inequalities are formulated as follows:

$$\boldsymbol{\zeta} + B\boldsymbol{h} = 0 \tag{6.21α′}$$

$$\Lambda^l_\zeta \boldsymbol{q} + \boldsymbol{b}^l_\zeta \leq \boldsymbol{\zeta} \leq \Lambda^u_\zeta \boldsymbol{q} + \boldsymbol{b}^u_\zeta \tag{6.21β′}$$

$$\boldsymbol{q}^l_{\text{ext}} \leq B^\top \boldsymbol{q} - \boldsymbol{q}_{\text{leak}} \leq \boldsymbol{q}^u_{\text{ext}} \tag{6.21γ′}$$

$$\Lambda^l_q \boldsymbol{h} + \boldsymbol{b}^l_q \leq \boldsymbol{q}_{\text{leak}} \leq \Lambda^u_q \boldsymbol{h} + \boldsymbol{b}^u_q \tag{6.21δ′}$$

$$\boldsymbol{g}^l \leq MB^\top \boldsymbol{q} - M\boldsymbol{q}_{\text{leak}} \leq \boldsymbol{g}^u \tag{6.21ε′}$$

$$\boldsymbol{v}^l_q \leq C_q \boldsymbol{q} - \hat{\boldsymbol{q}} \leq \boldsymbol{v}^u_q \tag{6.21στ′}$$

$$\boldsymbol{v}^l_p \leq C_h \boldsymbol{h} - \hat{\boldsymbol{p}} - \boldsymbol{z} \leq \boldsymbol{v}^u_p \tag{6.21ζ′}$$

$$\boldsymbol{q}^l \leq \boldsymbol{q} \leq \boldsymbol{q}^u \tag{6.21η′}$$

$$\boldsymbol{h}^l \leq \boldsymbol{h} \leq \boldsymbol{h}^u \tag{6.21θ′}$$

$$q^l_{\text{leak},j}\, y_j \leq q_{\text{leak},j} \leq q^u_{\text{leak},j}\, y_j \tag{6.21ι′}$$

$$\sum y_j = 1, j \in \mathcal{N} \tag{6.21ια′}$$

$$y_j \in \{0, 1\} \tag{6.21ιβ′}$$

where constraints (6.21ι′)-(6.21ιβ′) define the single leakage assumption. The linear constraints in (6.21) can be used to calculate a feasible set for the system states $\boldsymbol{x}$ using Linear Programming.

Because of the single leakage assumption, binary variables are introduced, which is undesirable considering the added computational complexity of solving a Mixed-Integer Linear Program (MILP), especially when dealing with large systems. In this work we propose the approximation of the set defined in (6.21ι′)-(6.21ιβ′) by the convex hull of this set, which is defined in the following lemma:

Lemma 1. The convex hull of the feasible set of $\boldsymbol{q}_{\text{leak}}$ given by (6.21ι′)-(6.21ιβ′), is given by the following linear constraints:

$$\sum_{j \in \mathcal{N}^u_0} \frac{q_{\text{leak},j}}{q^u_{\text{leak},j}} \leq 1 \tag{6.22α′}$$

$$q_{\text{leak},j} \geq 0 \;\forall j \in \mathcal{N} \tag{6.22β′}$$

$$q_{\text{leak},j} = 0 \;\forall j \in \mathcal{N} - \mathcal{N}^u_0 \tag{6.22γ′}$$

where $\mathcal{N}^u_0$ is the set of all nodes with a non-zero leakage upper bound, i.e., $\mathcal{N}^u_0 = \{j : q^u_{\text{leak},j} > 0, \forall j \in \mathcal{N}\}$.

Proof. Consider three variables, indicated by $q_i, i \in \{1, 2, 3\}$, out of which only one is non-zero, while if q_i is non-zero then $q_i \in \tilde{q}_i$ holds, such that $q^l_i \leq q_i \leq q^u_i$, $0 \leq q^l_i \leq q^u_i$, $\forall i$. The

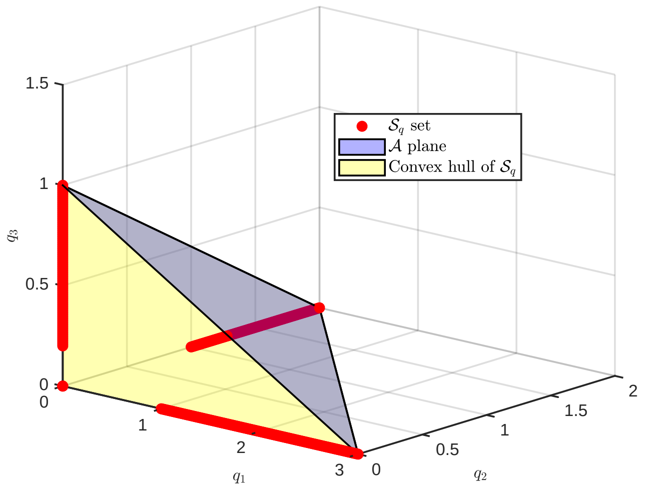


Figure 6.2. Convex hull of $\mathcal{S}_q$.

feasible set for the vector $\boldsymbol{q} = [q_1, q_2, q_3]^\top$ is then given by $\mathcal{S}_q$ as follows:

$$\mathcal{S}_q = \{q_1^l \leq q_1 \leq q_1^u, q_2 = 0, q_3 = 0\} \cup \tag{6.23}$$

$$\{q_1 = 0, q_2^l \leq q_2 \leq q_2^u, q_3 = 0\} \cup \tag{6.24}$$

$$\{q_1 = 0, q_2 = 0, q_3^l \leq q_3 \leq q_3^u\} \tag{6.25}$$

The convex hull of $\mathcal{S}_q$ can be defined as the intersection of the following half-spaces, as illustrated in Fig. 6.2:

$$\{q_1 \geq 0\} \cap \{q_2 \geq 0\} \cap \{q_3 \geq 0\} \cap \{\mathcal{A} \leq 0\} \tag{6.26}$$

where $\mathcal{A}$ is the plane passing through points $p_i = [q_{1i}, q_{2i}, q_{3i}]$, $i \in \{1, 2, 3\}$ explicitly defined as $p_1 = [q_1^u, 0, 0]$, $p_2 = [0, q_2^u, 0]$, $p_3 = [0, 0, q_3^u]$.

Let plane $\mathcal{A}$ be described by equation

$$aq_1 + bq_2 + cq_3 + d = 0 \tag{6.27}$$

where d is any non-zero number. Coefficients a, b, and c can be found by solving the following system of equations:

$$aq_{11} + bq_{21} + cq_{31} + d = 0 \tag{6.28α′}$$

$$aq_{12} + bq_{22} + cq_{32} + d = 0 \tag{6.28β′}$$

$$aq_{13} + bq_{23} + cq_{33} + d = 0 \tag{6.28γ′}$$

The solution to the system of equations (6.28) for the given points p_i yields:

$$a = \frac{-d}{q_1^u}, b = \frac{-d}{q_2^u}, c = \frac{-d}{q_3^u}. \tag{6.29}$$

Notice that the coefficients a, b, c depend only on their corresponding variable q_1, q_2, q_3 upper bound. Substituting the coefficients in (6.27), the equation describing plane $\mathcal{A}$ is given by

$$\frac{q_1}{q_1^u} + \frac{q_2}{q_2^u} + \frac{q_3}{q_3^u} = 1 \tag{6.30}$$

The half-plane $\mathcal{A} \leq 0$ is then given by:

$$\frac{q_1}{q_1^u} + \frac{q_2}{q_2^u} + \frac{q_3}{q_3^u} \leq 1 \tag{6.31}$$

It is then easy to generalize the constraints defining the convex hull of $\mathcal{S}_q$ for n dimensions, yielding:

$$q_i \geq 0, \forall i \in \{1, \ldots, n\} \tag{6.32}$$

$$\left(\sum_{i \in \{1,\ldots,n\}} \frac{q_i}{q_i^u} \right) \leq 1 \tag{6.33}$$

□

Two LPs for obtaining lower (**LPmin2**) and upper (**LPmax2**) bounds on each state variable $x_i \in \boldsymbol{z}$ are then formulated, as follows:

LPmin2:

$$\left\{ \begin{array}{ll} \min\limits_{\boldsymbol{z}} & x_i \\ s.t. & (6.21\alpha') - (6.21\theta'), \\ & (6.22) \end{array} \right\}$$

LPmax2:

$$\left\{ \begin{array}{ll} \max\limits_{\boldsymbol{z}} & x_i \\ s.t. & (6.21\alpha') - (6.21\theta'), \\ & (6.22) \end{array} \right\}$$

Similar to problems **LPmin** and **LPmax** in the detection phase, **LPmin2** and **LPmax2**, are a relaxed version of the original problem in (6.17), due to the bounding linearization of non-linearities, and moreover, the consideration in this formulation of every possible leakage location in the network. Using this approach it is ensured that the calculated state bounds for $x_i \in \left[x_i^l, x_i^u\right]$ include the true state value irrespective of where the leakage is. Problems **LPmin** and **LPmax** are solved for each state $x_i, i \in \mathcal{N}$, deriving bounds on the state vector $\boldsymbol{x}$. Note that state bounds are considerably more conservative compared to *Algorithm 1* due to the current constraints considering a leakage at every node.

In contrast, when calculating bounds on an individual leakage $q_{\text{leak},i}$, only the existence of the specific leakage should be considered. Leakage constraints are again modified as follows:

$$q_{\text{leak},i}^l \leq q_{\text{leak},i} \leq q_{\text{leak},i}^u, i \in \mathcal{N} \tag{6.34α′}$$

$$q_{\text{leak},l} = 0, \forall\, l \neq i \in \mathcal{N} \tag{6.34β′}$$

Due to the modified constraints, two separate LPs are formulated for obtaining lower (**LPmin2L**) and upper (**LPmax2L**) bounds on each leakage variable $q_{\text{leak},i} \in z$, as follows:

LPmin2L:

$$\left\{ \begin{array}{ll} \min\limits_{z} & q_{\text{leak},i} \\ s.t. & (6.21\alpha') - (6.21\theta') \\ & (6.34) \end{array} \right\}$$

LPmax2L:

$$\left\{ \begin{array}{ll} \max\limits_{z} & q_{\text{leak},i} \\ s.t. & (6.21\alpha') - (6.21\theta') \\ & (6.34) \end{array} \right\}$$

Leakage bounds are reduced by solving problems **LPmin2L** and **LPmax2L** for each leakage state $q_{\text{leak},i}, i \in \mathcal{N}$. **LPmin2L** or **LPmax2L** may be infeasible when it is impossible to find a solution under the consideration that all nodes are leakage-free except from node i. This signifies the impossibility of a leakage at node i; thus the leakage bounds are set to zero, such that $\tilde{q}_{\text{leak},i} = [0, 0]$.

The end of this procedure yields new bounds for the decision variable vector z. New bounds for the emitter coefficients $\tilde{c}$ are then calculated using (6.18).

6.4.4 Step 4 : Iterative reduction of leakage bounds

In this step, the calculated bounds on states and leakage are used to improve the problem constraints and iteratively reduce state and leakage bounds. This is achieved by reducing the feasible set at the next iteration through the procedure of bounding linearization. The iterations stop when the bounds on the vector of decision variables remain relatively unchanged. The convergence criterion used is that the change in bounds at iteration m, denoted by $e^{(m)}$ and defined as

$$e^{(m)} \triangleq \left| \left(z^{u\,(m)} - z^{l\,(m)} \right) - \left(z^{u\,(m-1)} - z^{l\,(m-1)} \right) \right|_1, \tag{6.35}$$

should be smaller than a small number ϵ, i.e., $e^{(m)} < \epsilon$.

6.4.5 Step 5: Considering multiple time-steps to improve localization

Assuming that the leakage persists until a known repair event, localization can be improved by exploiting a leakage characteristic that persists over multiple time-steps. Leakage magnitude q_{leak} varies between time-steps because of the varying pressure conditions in the network. In this work we exploit the fact that the leak hole area, which is proportional to the emitter coefficient c_j in (6.5), frequently remains constant during leakage events. Hence, information from previous time-steps can be passed onto the problem solved at time-step k

in the form of the previously calculated bounds on emitter coefficients. The initial bounds (iteration $m = 0$) on emitter coefficients at time-step k will then be $\tilde{\boldsymbol{c}}^{(0)}(k) = \tilde{\boldsymbol{c}}(k-1)$.

Algorithm 2 presents the complete procedure for leakage localization using an interval model over multiple time-steps.

Algorithm 2 Leakage localization algorithm

Input: time-step k, measurements $\hat{\boldsymbol{q}}(k)$, $\hat{\boldsymbol{p}}(k)$, demand bounds $\tilde{\boldsymbol{q}}_{\text{ext}}(k)$, $\tilde{\boldsymbol{g}}(k)$, parameter bounds $\tilde{r}$, emitter bounds $\tilde{\boldsymbol{c}}(k-1)$, max leak $\boldsymbol{q}_{\text{leak}}^{max}$

Output: $\tilde{\boldsymbol{q}}_{\text{leak}}(k)$, $\tilde{\boldsymbol{c}}(k)$

```
begin
Define leakage bounds $\tilde{\boldsymbol{q}}_{\text{leak}}^{(0)} = [0, \boldsymbol{q}_{\text{leak}}^{max}]$
Calculate state bounds $\tilde{\boldsymbol{x}}^{(0)}$
Define emitter bounds $\tilde{\boldsymbol{c}}^{(0)} = \tilde{\boldsymbol{c}}(k-1)$
Iteration: $m = 0$
while $e^{(m)} > \epsilon$ do
    Bounding linearization using $\tilde{\boldsymbol{z}}^{(m)}$, $\tilde{\boldsymbol{c}}^{(m)}$
    for $i = 1$ to $n_l + n_n$ do
        Solve LPmin2 and LPmax2 to find $x_i^l$ and $x_i^u$
    end for
    $\tilde{\boldsymbol{x}}^{(m+1)} = \left[\boldsymbol{x}^l,\ \boldsymbol{x}^u\right]$
    for $i = 1$ to $n_n$ do
        Solve LPmin2L and LPmax2L to find $q_{\text{leak},i}^l$ and $q_{\text{leak},i}^u$
        if infeasible then $\tilde{q}_{\text{leak},i} = [0, 0]$
        end if
    end for
    $\tilde{\boldsymbol{q}}_{\text{leak}}^{(m+1)} = \left[\boldsymbol{q}_{\text{leak}}^l,\ \boldsymbol{q}_{\text{leak}}^u\right]$
    Find $\tilde{\boldsymbol{c}}^{(m+1)}$ using $\tilde{\boldsymbol{q}}_{\text{leak}}^{(m+1)}$ and $\tilde{\boldsymbol{h}}^{(m+1)}$ in (6.18)
    $m = m + 1$
end while
$\tilde{\boldsymbol{q}}_{\text{leak}}(k) = \tilde{\boldsymbol{q}}_{\text{leak}}^{(m)}$, $\tilde{\boldsymbol{c}}(k) = \tilde{\boldsymbol{c}}^{(m)}$
return
```

6.5 Localization Priority Index

The proposed leakage localization procedure calculates bounds on node emitter coefficients, given by $\tilde{c}$. The nodes associated with a non-zero emitter coefficient upper bound, are the locations where the existence of a leak is feasible. We define the leakage-feasible node set $\mathcal{N}_l \subseteq \mathcal{N}$ as follows:

$$\{j \in \mathcal{N} : c_j^u > 0\} \in \mathcal{N}_l \,. \tag{6.36}$$

When all upper bounds c_j^u are zero except one, a single leak node is accurately identified. The leak node can be determined accurately when the leakage is large, in which case the leak node will have a non-zero lower bound, which confirms the leak location given that only one leakage exists in the system.

However, even when the leak location is singled out, additional information is included in the localization results in the form of the magnitude of the derived emitter coefficient bounds $\tilde{c}$. The comparison of node upper bounds can be used to prioritize possible leak locations, which can be a useful tool for water utility repair crews. In simple terms, *priority is given to nodes with larger emitter upper bounds*.

Previously proposed leakage localization assessment metrics in the literature typically depend on the localization results; e.g., when a single candidate leak node is identified, the topological distance from the actual leak is considered as the evaluation metric [155]. In general, most methodologies provide leakage hotspots, i.e., multiple candidate nodes, and the evaluation metric can be the average topological distance from the actual leak node [72], as well as the number of nodes contained in the leakage hotspot [156]. These evaluation approaches, in general, are not suitable for evaluating localization results in the form of a priority list of possible leak node locations, which may be useful for water utility repair crews.

In this work, in order to quantify how "good" the localization result is, we define the *Localization Priority Index* (LPI) as follows:

$$\text{LPI (\%)} = 100 \cdot \left(\sum_{i \in \mathcal{N} \neq l : c_i^u < c_l^u} 1 \right) / \, (n_n - 1) \,, \tag{6.37}$$

where l is the index of the leak node. The LPI essentially indicates the minimum percentage of nodes that are excluded from the node search space before the leakage is found by a leakage repair crew, given the priority defined by $\tilde{c}$. This is computed by finding the percentage of nodes with lower priority than the actual leak node l. The LPI $= 100\%$ when perfect localization is achieved, i.e., the leak node has the highest priority, while LPI $= 0\%$ when the leak node has the same or lower priority than all the other nodes.

6.6 Case studies

In this section we demonstrate the proposed leakage detection and localization methodology using benchmark case studies. In the first case study, a benchmark transport network is used to illustrate the results obtained by the proposed methodology when two specific leakages occur in the network. The second case study, uses a leakage diagnosis benchmark dataset containing sensor measurements from multiple operating scenarios of a benchmark network. Each scenario varies with respect to pipe parameters, water demands, leakage location and magnitude.

6.6.1 Example application on a transport network

The benchmark network 'Hanoi', shown in Fig. 6.3(a), is used to examine the performance of the proposed leakage diagnosis methodology. The 'Hanoi' network is a transport network with large demands at nodes of up to $300\ \mathrm{m}^3/\mathrm{hour}$. A version of the network, which is considered the "real" system, is simulated for 48 hours from time 00:00 hours, with a hydraulic time-step of $\Delta t = 30$ minutes. It contains unique realistic water demand patterns for each node with a 24 hour periodicity. A leakage occurs sometime in the first 24 hours, ensuring in this way that the performance of the algorithm is assessed under the daily varying network conditions.

A nominal hydraulic model of the network is given which is assumed topologically identical to the real network. The given pipe parameters are accurate within ±5% of the true parameter values. Moreover, demand pseudo-measurements are available for all nodes which are accurate within ±10% of the true demand values.

The real system is measured at the inlet of the network, i.e., the head at the reservoir node 1 and the flow at link 1. Additionally, 4 pressure sensors are installed at nodes $\{13, 16, 22, 30\}$ using a sensor placement procedure which maximizes the minimum sensitivity of all sensors to all possible leakages [77]. Pressure and flow sensor noise bounds $\tilde{\boldsymbol{v}}_p$ and $\tilde{\boldsymbol{v}}_q$ respectively are defined relative to the measurement value, such that $\tilde{\boldsymbol{v}}_p(k) = \pm 2\%\, \hat{\boldsymbol{p}}(k)$ and $\tilde{\boldsymbol{v}}_q(k) = \pm 2\%\, \hat{\boldsymbol{q}}(k)$.

A demand calibration methodology is considered to actively use the sensor measurements to give demand estimates for groups of nodes. To make the problem of calibration over-determined, the methodology divides the network into as many groups as the number of pressure sensors. Network partitioning is performed in this case by minimizing the flow exchange between node groups. Moreover, the following two constraints are imposed: i) one pressure sensor should belong in each group, ii) node groups should have the same number of nodes plus/minus two nodes. Algorithms that perform this kind of partitioning are available not only for water systems [157], but also for power systems [158] and smart buildings [159]. The calibration methodology is assumed to be able to give an estimate of a group demand within ±5% of the true group demand values [46].

In the first illustrative example of Fig. 6.3, an abrupt pressure-dependent leakage occurs at node $j = 32$ at 14:00 hours, with an emitter coefficient of $c_j = 17.3$ and an emitter exponent of $\alpha = 0.5$. This results in a leakage with magnitude of approximately 137 $\mathrm{m}^3/\mathrm{hour}$, which is 2.73% of the average system inflow for the given network. The effect of the leakage is unnoticeable by just observing the measurements, as illustrated in Fig. 6.3(b)–(c). In this example, the proposed leakage detection methodology detects the occurrence of a leakage immediately, i.e., given the first measurements that contain the effect of the fault.

After leakage detection, the localization procedure begins by calculating initial bounds on the leakage. These are assumed to be given by a methodology which monitors the inflow of the system. Here, conservative bounds of $\tilde{q}_{\mathrm{leak},j} = [50, 200]$, $\forall j \in \mathcal{N}$ are assigned. The localization procedure yields bounds on node emitter coefficients $\tilde{c}_j$, $\forall j \in \mathcal{N}$. The most useful information is provided by emitter upper bounds, as it can be used to define the priority of each node in the localization procedure. The emitter upper bounds derived in this example can be seen in Fig. 6.3(a). It is illustrated that nodes close to the leakage have a non-zero upper bound, indicating the feasibility of these nodes having a leakage given the uncertain model and measurements.

The priority given to nodes by their upper bounds is evaluated using the Localization Priority Index (LPI). The higher the LPI, the higher the priority given by the methodology to the node containing the leak. In Fig. 6.3(d), the LPI for the given example begins at 90% when the leakage is detected at 14:00 hours and increases to 100% by utilizing measurements from subsequent time-steps.

In the second illustrative example of Fig. 6.4, a smaller leakage occurs at the same node ($j = 32$) as in the first example at 16:00 hours, with an emitter coefficient of $c_j = 9.09$. This results in a leakage with average magnitude of 74.9 $\mathrm{m}^3/\mathrm{hour}$, which is approximately 1.53% of the average system inflow. In this example, the leakage is detected 6 hours after it occurs, at 22:00 hours. The initial value of LPI is 86.67% and gradually increases to 93.33% with the accumulation of new measurements, as seen in Fig 6.4(d). Due to the smaller leakage magnitude, it is feasible for the leak to exist in more nodes than in the previous example, as observed in Fig. 6.4(a). Interestingly though, the leak node is given top priority with only one

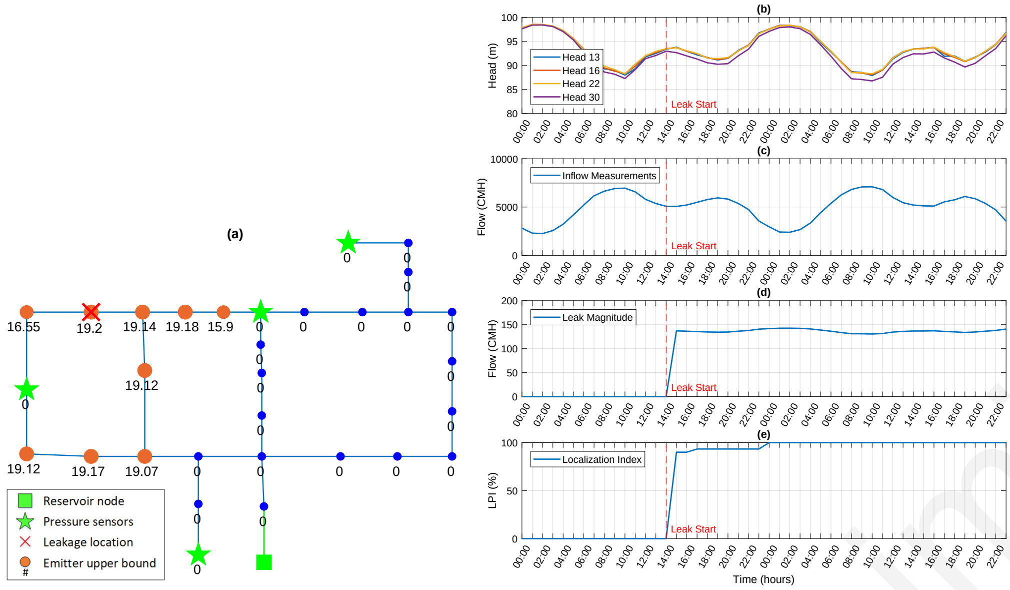


Figure 6.3. Example of leakage diagnosis on the Hanoi network: Leakage magnitude of ∼ 2.73% of average system inflow.

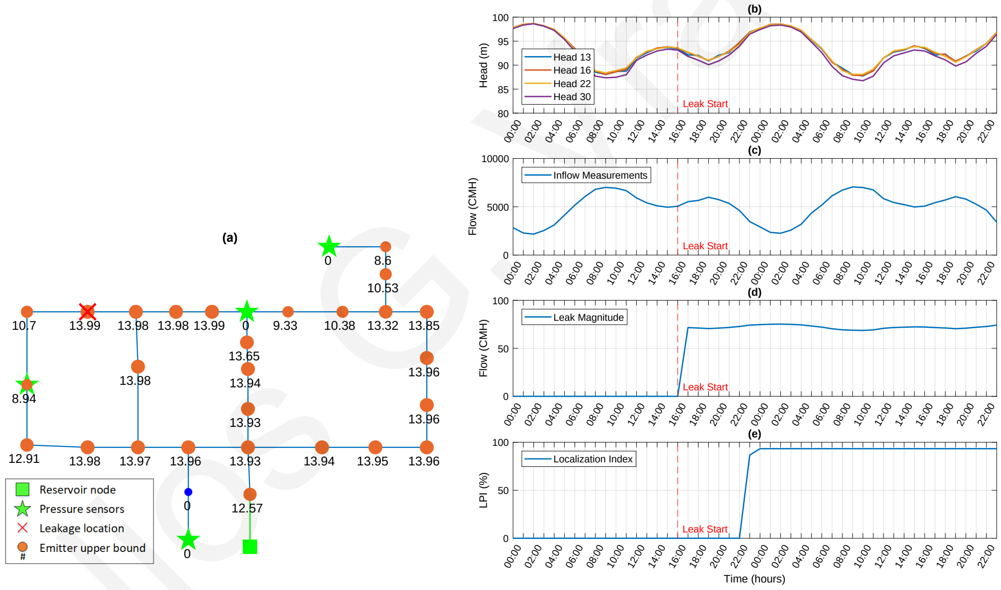


Figure 6.4. Example of leakage diagnosis on the Hanoi network: Leakage magnitude of ∼ 1.53% of average system inflow.

other node having the same emitter value, a result which translates in LPI = 93.33%

6.6.2 Methodology evaluation using LeakDB

In this subsection the proposed methodology is extensively tested using a benchmark dataset of leakages. The Leakage Diagnosis Benchmark (LeakDB) [11] is a realistic leakage dataset comprised of a large number of realistic leakage scenarios which occur randomly at

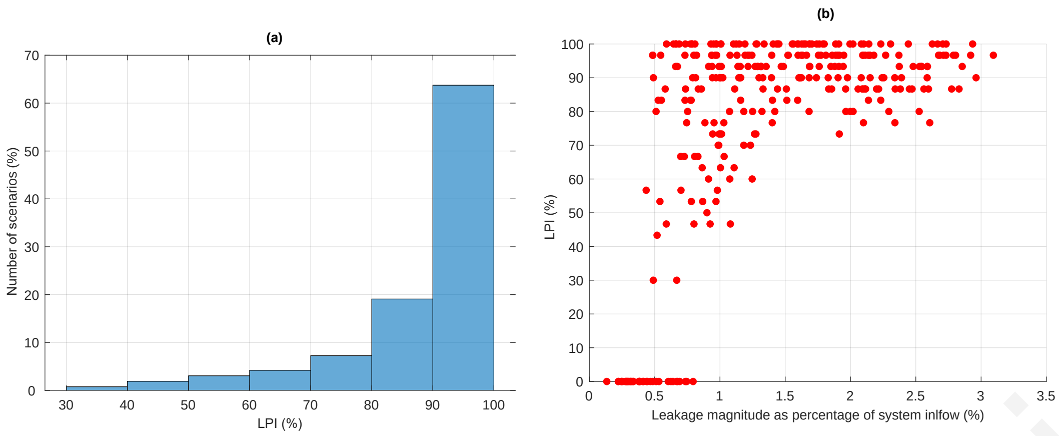


Figure 6.5. Application of the proposed Leakage Localization methodology on 500 scenarios: (a) Percentage of scenarios with specific range of Localization Priority Index value, (b) Localization Priority Index with respect to leakage magnitude

different water distribution benchmark networks, of different size and topology. For each benchmark network and for each leakage scenario, the leakage parameters (e.g., leak location and size), the structural parameters (e.g., length, pipe roughness) and realistic consumer pressure-driven demands are varied. For this case study, the dataset of the Hanoi benchmark network was selected.

In total, 500 different scenarios are used, in which the parameters and demands are varied randomly inside known uncertainty bounds. The demand uncertainty bounds $\tilde{\boldsymbol{q}}_{\text{ext}}(k)$ and parameter uncertainty upper bounds $\tilde{\boldsymbol{r}}$ are varied between ±10% and ±5% respectively of the given model nominal values. Moreover, a calibration methodology gives an estimate of group demands within ±5% of the true group demand values, given the sensor set.

It is assumed that flow and pressure are measured at the inlet of the network, i.e., the head at the reservoir Node 1 and the flow at Link 1. Additionally, 4 pressure sensors are installed at nodes $\{13, 16, 22, 30\}$ as in Examples 1 and 2. Pressure and flow sensor noise bounds $\tilde{\boldsymbol{v}}_p$ and $\tilde{\boldsymbol{v}}_q$ respectively are defined relative to the measurement value, such that $\tilde{\boldsymbol{v}}_p(k) = \pm 2\%\,\hat{\boldsymbol{p}}(k)$ and $\tilde{\boldsymbol{v}}_q(k) = \pm 2\%\,\hat{\boldsymbol{q}}(k)$.

In each scenario, a leakage is induced on a node, which is random in location and magnitude. It is possible for a scenario not to include a leakage. The leakages have a different profile, by varying the leak emitter coefficient. The emitter coefficients were calculated so that the resulting leakage magnitudes $q_{\text{leak}}(k)$ (m^3/h) are within $0-4\%$ of the average system inflow.

The benchmark also provides a scoring algorithm to evaluate the ability of an algorithm to detect leakages. Here the following standard metrics were extracted for the application of the

Table 6.1. Detection statistics from applying the proposed leakage detection algorithm based on Interval Model Invalidation (IMI) to a total of 500 leakage scenarios and comparison with the Interval State Estimation (ISE) based methodology proposed in Chapter 5.

Method	TP %	FN %	FP %	TN %	F1 %
IMI	87.33%	12.66%	0	100	93.24
ISE	56.28%	43.72%	0	100	72.02

proposed algorithm for leakage detection on all scenarios: 1) A fault existed and was detected (True Positive TP), 2) A fault existed but was not detected (False Negative FN), 3) A fault did not exist but a fault was detected (False Positive FP), 4) A fault did not exist and was not detected (True Negative TN), 5) The F1-score, a popular metric in datasets characterized by class imbalances [11]. The methodology in Chapter 5 [12], where interval-state estimation is used to create thresholds for leakage detection, is also applied on the same benchmark in order to compare its performance with the proposed model-invalidation methodology. The percentages for each case with 4 pressure sensors installed in the network are presented in Table 6.1.

The median and average detection delay for the scenarios where a detection occurs for the proposed methodology was 0 and 4.28 time-steps respectively. Both methodologies achieve zero false alarms when a fault is not present (0% False Positives and 100% True Negatives). This is due to the accurate knowledge of uncertainty bounds, which are provided by the LeakDB. As a result, the interval model is always valid when a fault does not exist.

It is worth noticing in Table 6.1 the high F1-score of the proposed methodology (IMI) compared to the methodology of Chapter 5 [12] (ISE). The improved performance is explained by the fact that when using model-invalidation the linear constraints formulated during the leakage detection procedure are part of the state-set representation, along with the bounds on states; in comparison, the state-set in the ISE based methodology is represented only by the bounds on states. This indicates that model invalidation is a more powerful tool than using detection thresholds created with models containing bounded uncertainties.

To evaluate the ability of the proposed methodology for leakage localization, the LPI in all scenarios was examined. In Fig. 6.5(a), scenarios with similar LPI are binned together, and the percentage of scenarios in a given LPI range is calculated. We can observe that the majority of scenarios have an LPI between $80-100\%$. Moreover, it is interesting to investigate the relationship between leakage magnitude and LPI. As seen in Fig. 6.5(a), the leakage scenarios with leakage magnitude below 1.5% of the average system inflow have a LPI which varies between $0-100\%$, depending on the leakage location. However, for leakages above 1.5%, the LPI remains above 70% in all cases.

6.7 Conclusions

In this work we proposed a methodology for leakage detection and localization in WDS using an interval-model, i.e., a hydraulic model in which uncertainties are modeled by intervals defined by a lower and upper bound. The interval-model is created using historical data of water demands by which uncertainty bounds are derived. Demand calibration data in the form of group demand estimates are also incorporated into the problem. Leakage detection is performed by incorporating real-time measurements from the network and checking the validity of the healthy interval-model. The results show increased detectability rates compared to similar methodologies which work using bounded uncertainties. Moreover, a methodology for leakage localization is proposed using this interval-model. The localization procedure yields a node priority list which can be used by water utilities to search for the exact leakage location using equipment based methods. Localization results on a leakage diagnosis benchmark dataset show that the leakage search space can be reduced to less than 20% of the network for leakage magnitudes above 1.5% of the system inflow.

Future work on the proposed approach will investigate the case when multiple leakages exist simultaneously in the network. The investigation of multiple-leakage detection and localization can be approached using a mixed-integer formulation with constraints on the number of simultaneous leakages. A mixed-integer formulation is more computationally expensive than the current proposed formulation, as discussed in Section 6.4.3, however, it is the first step to evaluate the performance of an extension of this methodology in the presence of multiple leakages, which considerably increase the feasible state set. As a note, the proposed methodology is designed to work in real-time to detect newly occurred leakages, while considering any existing leakages in the network to be included in the “healthy” system model. Consequently, it is realistic for the extension of this methodology not to consider a large number of simultaneous leakages.

Chapter 7

Bounded water-quality state estimation

The estimation of chlorine concentration in water distribution networks is a challenging task due to hydraulic and water-quality parameter uncertainties. In this chapter we propose a methodology for calculating chlorine concentration bounds at node locations of water distribution networks which is suitable for sensor fault and contamination detection purposes. The proposed *Backtracking Uncertainty Bounding Algorithm* (BUBA) considers known bounds on hydraulic states and water-quality model parameters to calculate the chlorine concentration bounds. The validity of the calculated bounds is demonstrated using a large number of Monte-Carlo simulations on benchmark networks. Moreover, the proposed methodology can be used in conjunction with a real-time parameter estimation algorithm, as it allows the use of time-varying bulk reaction coefficients. A parameter estimation algorithm is designed and implemented to approximate unknown bulk reaction coefficients of water originating from different sources. The BUBA then uses these time-varying parameters to improve the chlorine concentration bounds, as demonstrated in a case study with real data from a water transport network.

This chapter is organized as follows: Section 7.1 formulates the problem of modeling chlorine concentration in a network with uncertain flows and water quality parameters. Section 7.2 then describes the proposed Backtracking Uncertainty Bounding Algorithm. In Section 7.3 a parameter estimation algorithm which estimates bulk reaction coefficients in tanks is described. Illustrative results of applying the proposed BUBA on two benchmark networks are presented in Section 7.4.1 and 7.4.2. In section 7.4.3, results on a real network case study are shown, in which the BUBA is used in conjunction with the designed real-time parameter estimation algorithm.

Nomenclature

$\mathcal{G}$	directed graph of water network
$\mathcal{N}$	set of all nodes
$\mathcal{L}$	set of all links
$\mathcal{M}$	general model of water network
Θ	set of network parameters
n_n	number of nodes
n_l	number of links
h_i	head state at node i
q_{ij}	water flow state in link connecting node i to node j
$\boldsymbol{x}_h$	complete hydraulic-state of the network
Θ_h	set of hydraulic-model parameters
$\tilde{\boldsymbol{v}}$	interval vector notation, where $\underline{\boldsymbol{v}}$ is the lower bound vector and $\overline{\boldsymbol{v}}$ is the upper bound
c_i	chlorine concentration at node i
u_i	chlorine mass injection at node i
$\mathcal{U}$	set of nodes with known chlorine concentration
Θ_r	set of quality-model parameters
r^b	bulk reaction rate of chlorine in the network
r^w_{ij}	wall reaction rate of chlorine in pipe l_{ij}
$f_c(\cdot)$	nonlinear function for calculating chlorine concentration
m	memory of functions $f_c(\cdot)$ and $\tilde{f}_c(\cdot)$
t_d	detention time of water parcel in pipe
k_d	elapsed time steps of water parcel in pipe
x_d	auxiliary quantity defining the position of a water parcel in a pipe
α_{ij}	cross section area of pipe l_{ij}
λ_{ij}	length of pipe l_{ij}
c_{ij}	chlorine contribution of pipe l_{ij} to node j
$\mathcal{P}_j(k)$	set of all pipes that bring water into node j at time k
$\mathcal{T}_d$	set containing possible time delays
$\mathcal{N}_p$	set containing possible upstream nodes
$\mathcal{N}_s$	set of sensor nodes in the network
$v(k)$	tank water volume at time step k
α_t	tank cross section area
n_s	number of different water sources from which a tank receives water

7.1 Problem formulation

This work considers water distribution networks containing water tanks and other hydraulic elements such as pumps and valves. The topology of a WDS is modeled by a directed graph denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Let $\mathcal{N} = \{1, \cdots, n_n\}$ be the set of all nodes, where $|\mathcal{N}| = n_n$ is the total number of nodes. These represent junctions of pipes, consumer water demand locations, reservoirs and tanks. Let $\mathcal{L} = \left\{l_{ij} : i, j \in \mathcal{N}\right\}$ be the set of links such that link l_{ij} connects node i with node j, and $|\mathcal{L}| = n_l$ is the total number of links. These represent network pipes, water pumps and pipe valves, with the last two being the main hydraulic control elements in a water network. In addition, let $\mathcal{M}(\boldsymbol{\theta}, \mathcal{G})$ be the general model of a water distribution network which associates network parameters $\boldsymbol{\theta}$ with the network structure represented by the graph $\mathcal{G}$. Network parameters $\boldsymbol{\theta}$ are typically assumed time-invariant, except in the case of bulk reaction rates which may vary significantly.

7.1.1 Hydraulics modeling

The hydraulic-state associated with nodes is the *hydraulic head*, indicated by h_i, and is a specific measure of water pressure or water level above a geodetic datum. Each node $i \in \mathcal{N}$ is also associated with a water consumer demand at the node location, denoted by d_i. The unknown quantity associated with a link l_{ij} which connects node i to node j is the *water flow*, indicated by q_{ij}. Hydraulic network parameters $\boldsymbol{\theta}_h \subset \boldsymbol{\theta}$ represent parameters needed for hydraulic state estimation such as pipe lengths, diameters and roughness coefficients.

Hydraulic-state estimation is enabled using hydraulic-sensors in an observable configuration [47] which send measurements at a discrete time step k. A suitable hydraulic-state estimation algorithm then uses these measurements and the network model $\mathcal{M}(\boldsymbol{\theta}_h, \mathcal{G})$ to estimate the complete hydraulic-state given by $\boldsymbol{x}_h(k) = [\boldsymbol{h}(k), \boldsymbol{q}(k)]$, where $\boldsymbol{h} \in \mathbb{R}^{n_n}$ is the vector of heads at each node, and $\boldsymbol{q} \in \mathbb{R}^{n_l}$ is the vector of flows at each link.

When estimating hydraulic-states there are significant uncertainties in measurements, as well as model parameters. Measuring devices give readings which include measurement noise, causing the resulting reading to deviate from actual measured states, i.e. $\boldsymbol{y}_h(k) = C_h\boldsymbol{x}_h(k) + \boldsymbol{\nu}(k)$, where $\boldsymbol{y}_h$ are the measured hydraulic states, C_h is the output matrix and $\boldsymbol{\nu}$ is the measurement noise vector. Most importantly, estimated values of water demands (called pseudo-measurements) which may be needed for observability, are associated with significant uncertainty and lack statistical characterization of the error to the actual demands [40]. Parameter uncertainties include mainly pipe roughness coefficients, and the (less uncertain) pipe lengths and diameters.

In this work, the uncertainties are modeled as intervals, which is equivalent to a uniform probability distribution. For notational convenience, we adopt the convention of denoting intervals with a tilde. Let $\tilde{\boldsymbol{v}} = [\underline{\boldsymbol{v}}, \overline{\boldsymbol{v}}]$ be a closed interval vector, where $\underline{\boldsymbol{v}}$ is the lower bound vector and $\overline{\boldsymbol{v}}$ is the upper bound vector, such that: $\tilde{\boldsymbol{v}} = \{\boldsymbol{v} \in \mathbb{R}^n : \underline{v}_i \leq v_i \leq \overline{v}_i, \forall\, i = \{1, .., n\}\}$, and n is the size of the vector. We assume that bounds on model parameters and measurement noise exist and are available at each time step k. Bounds on hydraulic-states can then be calculated using the interval hydraulic-state estimator (IHISE) described in [2]. The interval-hydraulic-states are indicated by $\tilde{\boldsymbol{x}}_h(k) = [\tilde{\boldsymbol{h}}(k), \tilde{\boldsymbol{q}}(k)]$.

7.1.2 Water quality modeling

Water-quality-state in this work refers to the concentration of chlorine in the water at the nodes of the network. Chlorine concentration at a node i is indicated by $c_i(k)$, with k corresponding to the water-quality model discrete time step. Here for simplicity it is assumed that the quality model step duration Δt_q is equal to the hydraulic step duration Δt_h, i.e., $\Delta t_q = \Delta t_h = \Delta t$, and Δt is sufficiently small such that discretization errors are negligible.

Chlorine concentration is regulated in tanks using chlorine injection pumps, where the mass of chlorine injected in tank of node i is known and indicated by $u_i(k)$. Moreover, we assume that tanks have one distinct inlet and one outlet which are both monitored by chlorine sensors. Finally, tanks and locations in the network which have chlorine booster stations [160] are considered nodes belonging to set $\mathcal{U} \subset \mathcal{N}$, with known concentration.

The dynamics describing the change of chlorine concentration in the network are governed by: (1) the physical laws of chlorine transport, which depend on the water flows $\boldsymbol{q}(k)$ and the structure of the network $\mathcal{G}$, (2) the chemical laws of chlorine reactions, which depend on the decay rate of chlorine. Chlorine decay in turn depends on two factors: $(2a)$ the organic substances present in the water which result in the so-called *bulk reaction rate*, indicated by the time-dependent $r^b(k) < 0 \;\; \forall\, k$ and $(2b)$ the pipe l_{ij} the water flows in, which is characterized by the *pipe wall reaction rate*, indicated by $r^w_{ij} < 0$ [99]. Bulk and pipe wall reaction rates are the water-quality-model parameters of interest and are indicated by $\boldsymbol{\theta}_r(k) \subset \boldsymbol{\theta}$.

The exact chlorine decay dynamics in pipe networks are unknown, so modeling assumptions which approximate these dynamics are used [99]. Specifically, it is assumed that the chlorine decay in pipe networks can be described by a linear first order reaction model with varying reaction rate. Chlorine concentration at a node j is found by decaying the chlorine which originated from node i as follows:

$$c_j(t) = c_i\left(t - t_d\right) e^{r(t)\cdot t_d} + \eta(r), \tag{7.1}$$

where t_d is the time chlorine has spend in the pipe or detention time, $r(t) = r_{ij}^{w} + r^{b}(t)$ is the total decay rate in pipe l_{ij}, and $\eta(r)$ is an unknown function of the decay rate corresponding to modeling uncertainty. Notice that bulk decay rate $r^{b}(t)$ is assumed to be time-varying.

Moreover, it is assumed that there is instant and complete mixing of chlorine in pipe junctions. The concentration at a pipe junction j can be expressed as follows:

$$c_j(k) = \frac{\sum\limits_{i\in\mathcal{P}_j(k)} q_{ij}(k)c_{ij}(k)}{\sum\limits_{i\in\mathcal{P}_j(k)} q_{ij}(k)}, \tag{7.2}$$

where $c_{ij}(k)$ is defined as the chlorine contribution from a pipe l_{ij} to node j at time step k, and $\mathcal{P}_j(k)$ is the set of all pipes that bring water into node j at time k.

The chlorine concentration inside the network can be calculated using a water-quality estimator, which is a function that uses the physical and chemical laws that describe chlorine transport and reaction in water networks, to calculate the quality states. Let the nonlinear function $f_c(\cdot)$ be a water quality estimator to be defined, that calculates the chlorine concentration at the nodes of a network. The inputs and outputs of this function can be written as follows:

$$\boldsymbol{c}(k) = f_c\left(\boldsymbol{c}(0), \boldsymbol{U}_c(k_m), \boldsymbol{Q}(k_m); \mathcal{M}(\boldsymbol{\theta}_r; \mathcal{G})\right), \tag{7.3}$$

where:

- $\boldsymbol{c}(0)$ is the initial chlorine concentration at the nodes of the network,
- $\boldsymbol{U}_c(k_m) = [\boldsymbol{u}_c(k), \boldsymbol{u}_c(k-1), ..., \boldsymbol{u}_c(k-m)]$ is a sequence of chlorine injection control vectors,
- $\boldsymbol{Q}(k_m) = [\boldsymbol{q}(k), \boldsymbol{q}(k-1), ..., \boldsymbol{q}(k-m)]$ is a sequence of flow vectors,
- $m \in \mathbb{N}^{+}$ is the memory of function $f_c(\cdot)$, i.e., the maximum past time step of which the states and inputs are needed to evaluate this function.

7.1.3 Incorporating uncertainty in water-quality models

When estimating water quality, there is significant uncertainty due to uncertain water flows, uncertain wall reaction rates, as well as unknown and time-varying bulk reaction rates (due to the use of water from different sources). By modeling uncertain quantities as intervals, a chlorine concentration bounding estimator $\tilde{f}_c(\cdot)$ can be designed which is able to calculate bounds on chlorine concentration for a node i in the network, expressed as follows:

$$\tilde{c}_i(k) = \tilde{f}_c\Big(\boldsymbol{c}(0), \boldsymbol{U}_c(k_m), \widetilde{\boldsymbol{Q}}(k_m); \mathcal{M}(\widetilde{\Theta}_r(k_m); \mathcal{G})\Big), \tag{7.4}$$

where:

- $\tilde{c}_i(k) = \big[\underline{c}_i(k), \overline{c}_i(k)\big]$,
- $\tilde{\boldsymbol{Q}}(k_m)$ is a sequence of interval flow vectors, such that $\tilde{\boldsymbol{q}}(k) = \big[\underline{\boldsymbol{q}}(k), \overline{\boldsymbol{q}}(k)\big]$,
- $\tilde{\Theta}_r(k_m)$ is a sequence of interval parameter vectors, such that $\tilde{\boldsymbol{\theta}}_r(k) = \big[\underline{\boldsymbol{\theta}}_r(k), \overline{\boldsymbol{\theta}}_r(k)\big]$.

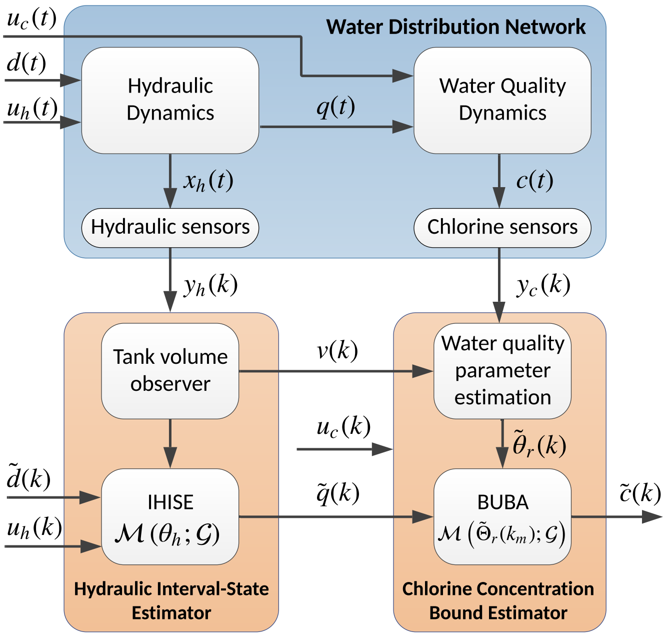
The aim of this work is to design a methodology which realises the function $\tilde{f}_c(\cdot)$ in (7.4).

7.2 The Backtracking Uncertainty Bounding Algorithm

A chlorine concentration bounding estimator $\tilde{f}_c(\cdot)$ is realised in this section by introducing the *Backtracking Uncertainty Bounding Algorithm* (BUBA). This methodology uses interval hydraulic-state estimates, specifically water flows $\tilde{\boldsymbol{q}}(k)$, from the IHISE algorithm described in [2] as input. It can also use time-varying water-quality parameters $\boldsymbol{\theta}_r(k)$, specifically bulk reaction rate estimates which are updated online using a suitable parameter estimation algorithm. Chlorine sensor measurements $\boldsymbol{y}_c(k)$ can then be compared with the chlorine concentration bounding estimates $\tilde{\boldsymbol{c}}(k)$ calculated by the BUBA and detect contamination events using a suitable contamination detection logic. A general diagram illustrating this process is shown in Fig. 7.1.

7.2.1 The backtracking approach

In the backtracking approach [102, 101], chlorine concentration of *water parcels* arriving at sensor locations are back-tracked throughout the network until boundary conditions are reached. These boundary conditions are: (a) Space boundary, which is the case when the parcel reaches an input node in $\mathcal{U}$ of which the concentration is known; (b) Time boundary, which is the case when the water parcel reaches the initial time step $k = 0$ at which the chlorine concentration $\boldsymbol{c}(0)$ of all nodes is known. Note that if the simulation is executed for a sufficient time, any chlorine initially present in the network will have decayed. Thereafter, the output concentration will only depend on chlorine originating from input nodes.



Notation
d: nodal water demands
q: pipe water flows
v: tank volume
u_h: hydraulic inputs
y_h: hydraulic sensor measurements
x_h: hydraulic states
c: nodal chlorine concentration
θ_r: water-quality parameters
u_c: chlorine injection
y_c: chlorine concentration measurements

Figure 7.1. Information flow diagram for using the proposed methodology for contamination detection.

7.2.2 The BUBA functions

The BUBA consists of four functions that recursively call each other to collectively compute the chlorine concentration at a node. The first function called is *Function 4* which checks the boundary conditions for a given node j and time step k. If boundary conditions have not been reached, *Function 3* is called to calculate the concentration and bulk reaction rate at node j considering this is a pipe junction. To do this, *Function 3* calls *Function 2* which provides the chlorine contribution and bulk reaction rates of each pipe connected to the junction. In turn, *Function 2* calls *Function 1* to provide the upstream node and detention times for each pipe, while it also recursively calls *Function 4* to provide the chlorine concentration at the upstream node at a previous time step. An illustration of the recursive calls and information flow of the BUBA functions is shown in Fig. 7.2. A description of each function follows.

Function 1 — Detention time in single pipe with uncertainty

Here the methodology for finding the detention time and upstream node of a water parcel moving in a single pipe under uncertain water flow is described.

Consider the simple case of a single pipe, as in Fig. 7.3, of which the output node j receives water from the input node i. Chlorine transport is expressed by finding the time the

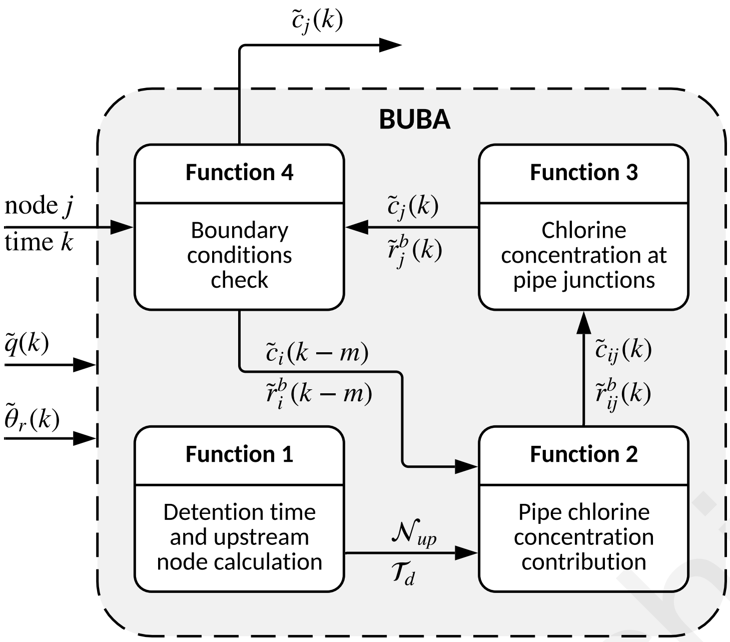


Figure 7.2. Information flow between the BUBA functions

water parcel has spend in the pipe, or detention time, indicated by $t_d \in \mathbb{R}^+_0$. The detention time is calculated by finding the number of elapsed time steps $k_d \in \mathbb{N}^+$, or delay, since the water parcel currently arriving at node j has entered pipe l_{ij}. Note that the upstream node $p \in \mathcal{N}$, defined as the node from which the parcel originated from, should be determined since this node can be either i or j due to flow reversal. The auxiliary quantity x_d defines the position of the water parcel in the pipe, using the water flow q_{ij} and the pipe cross section area α_{ij} as follows:

$$x_d = \frac{\Delta t}{\alpha_{ij}} \sum_{m=1}^{k_d} q_{ij}(k-m). \tag{7.5}$$

The delay k_d and the upstream node p is calculated by finding the minimum value of k_d that results in: (i) x_d being larger than the length of the pipe λ_{ij}, in which case the resulting upstream node $p = i$, or (ii) x_d being less than zero, in which case the upstream node $p = j$:

$$k^*_d = \begin{cases} \min & \mathrm{k_d} \in \mathbb{N}^+ \\ s.t. & x_d \notin \left[0, \lambda_{ij}\right] \end{cases} \tag{7.6}$$

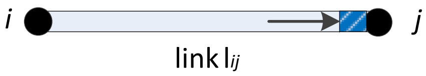
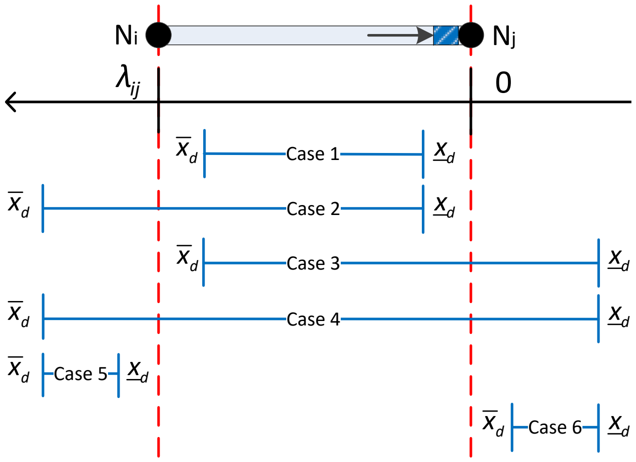


Figure 7.3. Water parcel arriving at node j through pipe l_{ij}



Case 1: $0 < \left[\underline{x}_d, \overline{x}_d\right] < L_{ij}$

Case 2: $0 < \underline{x}_d < L_{ij},\, \overline{x}_d > L_{ij}$

Case 3: $\underline{x}_d < 0,\, \overline{x}_d > L_{ij}$

Case 4: $\underline{x}_d < 0,\, \overline{x}_d > L_{ij}$

Case 5: $\left[\underline{x}_d, \overline{x}_d\right] > L_{ij}$

Case 6: $\left[\underline{x}_d, \overline{x}_d\right] < 0$

Figure 7.4. The interval $\tilde{x}_d$ with respect to the pipe length

The exact detention time t_d is then calculated as follows:

$$t_d = \left(k_d^* - 1\right)\Delta t + \frac{\alpha_{ij}\left(\lambda_{ij} - x_d\right)}{q_{ij}(k - k_d^*)}. \tag{7.7}$$

Consider now the single pipe of Fig. 7.4 in which a water parcel moves under the effect of the uncertain water flow $\tilde{q}_{ij}$. Since we have an interval of possible flow values, the variable x_d will also be an interval. The interval of possible positions for the water parcel $\tilde{x}_d \in \left[\underline{x}_d, \overline{x}_d\right]$ can be calculated as follows:

$$\left[\underline{x}_d, \overline{x}_d\right] = \frac{\Delta t}{\alpha_{ij}} \sum_{m=1}^{k_d} \left[\underline{q}_{ij}(k-m), \overline{q}_{ij}(k-m)\right]. \tag{7.8}$$

Using interval arithmetic [133], we can express (7.8) as:

$$\left[\underline{x}_d, \overline{x}_d\right] = \frac{\Delta t}{\alpha_{ij}} \left[\sum_{m=1}^{k_d} \underline{q}_{ij}(k-m),\ \sum_{m=1}^{k_d} \overline{q}_{ij}(k-m)\right]. \tag{7.9}$$

The interval $\left[\underline{x}_d, \overline{x}_d\right]$ can reside in one of six positions, with respect to the pipe length, as illustrated in Fig. 7.4. The interval of possible positions $\left[\underline{x}_d, \overline{x}_d\right]$ is used to deduce the detention time and upstream nodes. Because this is an interval, there will be many possible detention times that may correspond to different upstream nodes. In (7.9) we gradually increase delay k_d until: (i) both $\underline{x}_d$ and $\overline{x}_d$ are outside the pipe range, i.e., $\left[\underline{x}_d, \overline{x}_d\right] \not\subset \left[0, \lambda_{ij}\right]$ or (ii) the time boundary is reached, i.e., $k_d = k$. For each iteration that either $\underline{x}_d$ or $\overline{x}_d$ are outside the pipe range, the detention time t_d is calculated using (7.7) and saved along with the corresponding upstream node. When evaluating (7.7), the lower bound of flows should be used if $\underline{x}_d$ is outside the pipe range, and the upper bound of flows should be used if $\overline{x}_d$ is outside the pipe range. This procedure calculates two sets: $\mathcal{T}_d$ containing the possible time delays and $\mathcal{N}_p$ containing the possible upstream nodes corresponding to each time delay, i.e., $\mathcal{T}_d \rightarrow \mathcal{N}_p$.

Function 1 Water detention time range in a pipe and set of upstream nodes

```
Input: node $j$, pipe $l_{ij}$, time step $k$
begin
  $k_d \leftarrow 1$
  repeat
      Calculate $\left[\underline{x}_d, \overline{x}_d\right]$ using (7.9)
      Case 1 : $k_d \leftarrow k_d + 1$
      Case 2 – 6 : Calculate delay $t_d$ and node $p$
                  Add in $\mathcal{T}_d$ and $\mathcal{N}_p$ respectively
                  $k_d \leftarrow k_d + 1$
  until $\left(\left[\underline{x}_d, \overline{x}_d\right] \not\subset \left[0, \lambda_{ij}\right]\right) \vee (k_d = k)$
return $\mathcal{T}_d, \mathcal{N}_p$
```

Function 2 — Chlorine and bulk reaction rate contribution of a pipe to a node

The methodology for finding the interval of possible chlorine contributions $\tilde{c}_{ij}(k) = [\underline{c}_{ij}(k), \overline{c}_{ij}(k)]$, of a single pipe l_{ij} to a node j (as in Fig. 7.3) is described and implemented in *Function 2*. The function also finds the bounds on the possible bulk reaction rate contribution $\tilde{r}^b_{ij}(k) = [\underline{r}^b_{ij}(k), \overline{r}^b_{ij}(k)]$ of pipe l_{ij} to node j.

Uncertainty on chlorine decay rates is considered in this function for the calculation of chlorine concentration: the uncertain wall reaction rate of the specific pipe given by $\tilde{r}^w_{ij} = [\underline{r}^w_{ij}, \overline{r}^w_{ij}]$ and the uncertain and time-varying bulk reaction rate of the upstream node given by $\tilde{r}^b_p(k) = [\underline{r}^b_p(k), \overline{r}^b_p(k)]$. The upstream node chlorine concentration is also uncertain given by the interval $\tilde{c}_p(k) = [\underline{c}_p(k), \overline{c}_p(k)]$. The interval of uncertain pipe wall reaction rates $\tilde{r}^w_{ij}$ is assumed known, derived from knowledge of pipe material. The interval states $\tilde{r}^b_p(k)$ and $\tilde{c}_p(k)$ are calculated recursively by *Function 4*. Considering the availability of sets $\mathcal{T}_d$ and $\mathcal{N}_p$, given by *Function 1*, the chlorine contribution bounds of pipe l_{ij} to node j is found as follows:

$$\underline{c}_{ij}(k) = \begin{cases} \min\limits_{m} & e^{\left(\tilde{r}^w_{ij}+\tilde{r}^b_i(k-m)\right)\Delta t\, m} \cdot \tilde{c}_i\left(k-m\right) \\ s.t. & m \in \mathcal{T}_d \rightarrow i \in \mathcal{N}_p \end{cases}, \tag{7.10}$$

while $\overline{c}_{ij}(k)$ is found by solving (7.10) as a maximization problem. These problems are solved using interval arithmetic and enumeration of the possible time steps $m \in \mathcal{T}_d$. Similarly, the bounds on possible bulk reaction rate contribution of pipe l_{ij} to node j is found by solving

Function 2 Pipe chlorine concentration and bulk reaction rate contribution bounds

```
Input: node $j$, pipe $l_{ij}$, time step $k$
begin
    Get $\mathcal{N}_p$ and $\mathcal{T}_d$ by calling Function 1
    for $m \in \mathcal{T}_d \to i \in \mathcal{N}_p$ do
        Get $\tilde{c}_i(k-m)$ and $\tilde{r}_i^b(k-m)$ by calling Function 4
    end for
    Solve (7.10) to get $\tilde{c}_{ij}(k)$.
    Solve (7.11) to get $\tilde{r}_{ij}^b(k)$.
return $\tilde{c}_{ij}(k), \tilde{r}_{ij}^b(k)$
```

the following problem:

$$\underline{r}_{ij}^b(k) = \begin{cases} \min\limits_{m} & \tilde{r}_i^b\,(k-m) \\ s.t. & m \in \mathcal{T}_d \to i \in \mathcal{N}_p \end{cases}, \tag{7.11}$$

while $\bar{r}_{ij}^b(k)$ is found by solving (7.11) as a maximization problem.

Function 3 — Chlorine concentration and bulk reaction rate at pipe junctions

This section describes the methodology for finding the chlorine concentration and bulk reaction rate bounds of a junction node that receives water from multiple pipes. The methodology is implemented in *Function 3*.

This function utilizes Assumption 7.1.2, which dictates that when multiple pipes bring water into a node the water and chlorine in the water mix instantly and completely. In the case when no uncertainty is considered, this is described by (7.2). In the case when uncertainties are present, all quantities in (7.2) will be represented by intervals. The interval of possible values for chlorine concentration at node j, indicated by $\tilde{c}_j(k) = [\underline{c}_j(k), \bar{c}_j(k)]$, is calculated by solving the following optimization problem:

$$\underline{c}_j(k) = \begin{cases} \min\limits_{q_{ij}} & \dfrac{\sum\limits_{i\in\mathcal{P}_j(k)} q_{ij}(k)\,\left[\underline{c}_{ij}(k),\bar{c}_{ij}(k)\right]}{\sum\limits_{i\in\mathcal{P}_j(k)} q_{ij}(k)} \\ s.t. & \underline{q}_{ij}(k) < q_{ij}(k) < \bar{q}_{ij}(k) \end{cases}, \tag{7.12}$$

while $\bar{c}_j(k)$ is found by solving (7.12) as a maximization problem. The interval $\left[\underline{c}_{ij}(k), \bar{c}_{ij}(k)\right]$ is given by *Function 2*. Problem (7.12) is a linear fractional problem with interval coefficients which can be solved using the formulation described in [161].

Function 3 Chlorine concentration and bulk reaction rate bounds at pipe junctions

```
Input: node $j$, time step $k$
begin
    Find $\mathcal{P}_j(k)$
    for $i \in \mathcal{P}_j(k)$ do
        Find $\tilde{c}_{ij}(k)$ and $\tilde{r}^b_{ij}(k)$ by calling Function 2
    end for
    Solve (7.12) to get $\tilde{c}_j(k)$
    Solve (7.13) to get $\tilde{r}^b_j(k)$
return $\tilde{c}_j(k), \tilde{r}^b_j(k)$
```

The bulk reaction rate of the water arriving at a node of interest is calculated in a similar manner. Assumption 7.1.2 is used which implies that bulk reaction rate is proportional to the volume of water mixing from each water source. The interval of bulk reaction rates at node j, indicated by $\tilde{r}^b_j(k) = [\underline{r}^b_j(k), \overline{r}^b_j(k)]$, is calculated as follows:

$$\underline{r}^b_j(k) = \begin{cases} \min\limits_{q_{ij}} & \dfrac{\sum\limits_{i \in \mathcal{P}_j(k)} q_{ij}(k) \left[\underline{r}^b_{ij}(k), \overline{r}^b_{ij}(k)\right]}{\sum\limits_{i \in \mathcal{P}_j(k)} q_{ij}(k)} \\ s.t. & \underline{q}_{ij}(k) < q_{ij}(k) < \overline{q}_{ij}(k) \end{cases}, \tag{7.13}$$

while $\overline{r}^b_j(k)$ is found by solving (7.13) as a maximization problem.

Function 4 — Chlorine concentration and bulk reaction rate bounds at a node

Function 4 is the top-level function of the proposed methodology, which is called to calculate the chlorine concentration bounds of a general node j. The function checks if the boundary conditions have been reached, i.e., if the initial time step $k = 0$ has been reached (time boundary) or if the node of interest is an input node (space boundary). If any of the above is true then the chlorine concentration and bulk reaction rate of the node is known, otherwise, *Function 3* is called to calculate the chlorine concentration and bulk reaction rate bounds of node j.

7.3 Real-time chlorine decay rate estimation in tanks

In this section, the methodology for estimating the chlorine decay rate inside a water tank which receives water from different sources is described. In many water transport net-

Function 4 Chlorine concentration and bulk reaction rate bounds of a general node

```
Input: node $j$, time step $k$
begin
if $k = 0$ then
    return $\tilde{c}_j(0), \tilde{r}^b_j(0)$
end if
if $j \in \mathcal{U}$ then
    return $\tilde{c}_j(k), \tilde{r}^b_j(k)$
end if
Find $\tilde{c}_j(k)$ and $\tilde{r}^b_j(k)$ by calling Function 3
return $\tilde{c}_j(k), \tilde{r}^b_j(k)$
```

works, water from various sources is first stored in tanks and then distributed to the rest of the network. Water supplied from different sources may have varying quality characteristics; i.e. contains different concentrations of organic matter, which in turn affects chlorine decay rates. The quality characteristics of each source may be unknown and so would be the corresponding chlorine decay rate parameters.

In this work, we assume that hydraulic sensors are used to monitor the tank hydraulic-states. Moreover, chlorine concentration sensors are installed at the input and output of the tanks. Under these conditions, a chlorine bulk reaction rate estimation scheme is designed to estimate in real-time the changing characteristic of water in tanks. The time-varying bulk reaction rate estimation algorithm is then used in conjunction with the BUBA to improve the chlorine concentration bounding estimates in the network.

7.3.1 Chlorine estimation in tanks

The model used in this work for calculating chlorine concentration in a tank which receives water from multiple water sources makes use of the following assumptions:

- There is instant and complete mixing of water and chlorine in the tank.
- Chlorine decay in tanks can be approximated by a linear, first order reaction model with time-varying decay rate.
- Water from each source is characterized by a distinct bulk reaction rate $r_i(k)$.
- The effect of tank wall reactions r^w is negligible compared to tank bulk reactions.

Considering the aforementioned assumptions, a chlorine concentration model for tanks is given by:

$$c_{out}(k+1) = \frac{v(k) - \Delta t\, q_{out}(k) + \Delta t \sum_{i=1}^{n_s} \left(v_i(k)\, r_i(k)\right)}{v(k+1)} c_{out}(k) + \frac{\Delta t \sum_{i=1}^{n_s} \left(q_{in,i}(k)\, c_{in,i}(k)\right) + u_c(k)}{v(k+1)}, \quad (7.14)$$

where:

- $c_{out}(k)$ is the chlorine concentration at the tank output at time step k, which is equal to the concentration inside the tank,
- $\boldsymbol{c}_{in}(k) \in \mathbb{R}^{n_s}$ is the vector of chlorine concentrations in the water of each of the n_s different sources,
- $u_c(k) \in \mathbb{R}$ is the chlorine mass input inside the tank,
- $\boldsymbol{q}_{in}(k) \in \mathbb{R}^{n_s}$ is the vector of tank inflows from n_s different sources,
- $q_{out}(k) \in \mathbb{R}$ is the tank outflow,
- $v(k) = h_t(k)\, \alpha_t = \sum_{i=1}^{n_s} v_i(k)$ is the volume of water inside the tank,
- $h_t(k) \in \mathbb{R}$ is the tank water level,
- α_t is the tank base area,
- $v_i(k)$ is the volume of water in the tank from source i at time k and
- $r_i(k)$ is the unknown and time-varying chlorine decay rate of water from source i.

All the quantities in (7.14) needed to calculate c_{out} are either known or measured, except the vector of chlorine decay rates from each source. The chlorine decay rate vector is denoted as $\boldsymbol{r}(k) \in [r_1(k)\, r_2(k)\, \cdots\, r_{n_s}(k)]$.

7.3.2 Monitoring water volume from different sources

The hydraulic-states corresponding to tank water levels can be estimated using a discrete-time Luenberger observer, as shown in [12]. Water volumes inside the tank can be calculated using these water levels and the structure of the tank. Here, an extension of the discrete-time observer described in [12] is proposed, able to estimate the water volume from each source inside a tank. First, an estimate of the total tank volume $\hat{v}(k)$ is calculated using the following observer structure:

$$\hat{v}(k+1) = \hat{v}(k) + \Delta t \sum_{i=1}^{n_s} q_{in,i}(k) - \Delta t\, q_{out}(k) + L_d \left(h_t(k)\, \alpha - \hat{v}(k)\right), \quad (7.15)$$

where $L_d \in (0,1)$ is the discrete-time observer gain. To estimate the volume from each source i in the tank, the input $q_{in,i}$ and output $q_{out,i}$ from each source is needed. The input from each source is typically measured, however the output cannot be measured after the water is mixed in the tank. Using the assumption of complete mixing of water in the tank, it can also be assumed that the volume of water from each source exiting the tank at each time instant is proportional to the water volume from this source inside the tank, such that:

$$q_{out,i}(k) = v_i(k)/v(k) \cdot q_{out}(k). \tag{7.16}$$

Given (7.16), the water volume $\hat{v}_i(k)$ from each source i is estimated using $\hat{v}(k)$ as follows:

$$\hat{v}_i(k+1) = \hat{v}_i(k) + \Delta t\, q_{in,i}(k) - \Delta t\, q_{out,i}(k) = \hat{v}_i(k) + \Delta t\, q_{in,i}(k) - \Delta t\, \frac{\hat{v}_i(k)}{\hat{v}(k)} q_{out}(k). \tag{7.17}$$

Note that when calculating (7.17), mass conservation must be taken into account, i.e., the total water volume in the tank should be equal to the sum of individual water volumes originating from each source, as follows:

$$\hat{v}(k+1) = \sum\nolimits_{i=1}^{n_s} \hat{v}_i(k+1). \tag{7.18}$$

7.3.3 Initial decay rate estimates

The actual decay rates $\boldsymbol{r}(k)$ in (7.14) are unknown, however initial estimates $\hat{\boldsymbol{r}}(0)$ can be derived using historical data as a training set. The initial estimates may improve the performance of the parameter estimation algorithm if they are close to the actual parameter values. In this work, the initial estimates are mainly used to compare the performance of chlorine estimation when using constant and time-varying decay rates. The constant initial estimates are calculated using k_T chlorine measurements $c_{out}(k)$, $k \in \{-k_T, \cdots, -1\}$ at the tank output as a training set. The chlorine estimation function of (7.14) is modified to consider constant decay rates $\boldsymbol{r}$ and, in order to use already measured quantities, time delay is added, as follows:

$$\hat{c}_{out}(k, \boldsymbol{r}) = \frac{\hat{v}(k-1) - \Delta t\, q_{out}(k-1) + \Delta t \sum\limits_{i=1}^{n_s} \hat{v}_i(k-1)\, r_i}{\hat{v}(k)} \hat{c}_{out}(k-1) + \frac{\Delta t \sum\limits_{i=1}^{n_s} q_{in,i}(k-1)\, c_{in,i}(k-1) + u_c(k-1)}{\hat{v}(k)}, \tag{7.19}$$

where $\hat{c}_{out}(k, \boldsymbol{r})$ is the estimation of chlorine concentration at time k using the constant decay parameters $\boldsymbol{r}$. The initial estimates $\boldsymbol{r}(0)$ are calculated by solving the following least squares optimization problem:

$$\hat{\boldsymbol{r}}(0) = \begin{cases} \arg\min\limits_{\boldsymbol{r}} & \sum\limits_{k=-k_T}^{-1} \left[c_{out}(k) - \hat{c}_{out}(k, \boldsymbol{r})\right]^2 \\ s.t. & \underline{\boldsymbol{r}} \leqslant \boldsymbol{r} \leqslant \bar{\boldsymbol{r}} \end{cases}, \tag{7.20}$$

where $\underline{\boldsymbol{r}}$ and $\bar{\boldsymbol{r}}$ are vectors of known, conservative, lower and upper bounds on the decay rates respectively.

7.3.4 Real-time parameter estimation scheme

Having the chlorine dynamics model with known parameters of (7.14) and an initial estimate of chlorine decay rates $\hat{\boldsymbol{r}}(0)$, a real-time parameter estimation scheme for each source's decay rate can be designed. The chlorine concentration model of (7.14) is re-written by adding time-delay and by separating the measured (known) terms $z(k), \boldsymbol{\psi}(k)$ and the unknown terms $\boldsymbol{r}(k)$. To do this, we first define the known terms $z(k)$ as follows:

$$\begin{aligned} z(k) &\triangleq \hat{v}(k) \cdot c_{out}(k) - \hat{v}(k-1) \cdot c_{out}(k-1) \\ &- \Delta t \sum_{i=1}^{n_s} q_{in,i}(k-1)\, c_{in,i}(k-1) + \Delta t \cdot q_{out}(k-1) \cdot c_{out}(k-1) - u_c(k-1). \end{aligned} \tag{7.21}$$

The time-delayed version of the chlorine concentration model (7.14) is then re-written using $z(k)$ as follows:

$$z(k) = \underbrace{\begin{bmatrix} r_1(k-1) \\ \vdots \\ r_{n_s}(k-1) \end{bmatrix}^{\top}}_{\boldsymbol{r}(k-1)} \underbrace{\begin{bmatrix} v_1(k-1) \\ \vdots \\ v_{n_s}(k-1) \end{bmatrix} \Delta t \; c_{out}(k-1)}_{\boldsymbol{\psi}(k-1)} \tag{7.22}$$

The actual decay rate parameters $\boldsymbol{r}(k)$ can be expressed using $z(k)$ and $\boldsymbol{\psi}(k-1)$ as follows:

$$z(k) = \boldsymbol{r}^{\top}(k-1)\boldsymbol{\psi}(k-1), \tag{7.23}$$

while the estimated parameters, denoted by $\hat{\boldsymbol{r}}(k)$, are given by:

$$\hat{z}(k) \triangleq \hat{\boldsymbol{r}}(k-1)\boldsymbol{\psi}(k-1) \tag{7.24}$$

The error between the measured and estimated quantities is then calculated as follows:

$$\varepsilon(k) = \frac{z(k) - \hat{z}(k)}{m^2(k-1)} \tag{7.25}$$

where $m(k-1) > 0$ a normalized signal which bounds $\boldsymbol{\psi}(k-1)$ from above, e.g., $m^2(k-1) = 1 + \boldsymbol{\psi}^{\top}(k-1)\boldsymbol{\psi}(k-1)$. By performing Lyapunov analysis we get the algorithm to calculate the estimate $\hat{\boldsymbol{r}}(k)$ [162]:

$$\begin{aligned} \varepsilon(k) &= \frac{z(k) - \hat{\boldsymbol{r}}^{\top}(k-1)\boldsymbol{\psi}(k-1)}{m^2(k-1)} \\ \hat{\boldsymbol{r}}(k) &= \hat{\boldsymbol{r}}(k-1) + \Gamma\varepsilon(k)\boldsymbol{\psi}(k-1) \end{aligned} \tag{7.26}$$

where $\Gamma \in (0,2)$ is a design parameter which affects how fast the parameter estimates change. As a practical note, the estimated decay rates from different sources are not expected to vary significantly in the course of a day, thus parameter Γ should be chosen to be relatively small.

Finally, an estimate of the overall chlorine decay rate at tank outflow $\hat{r}_{out}(k)$ is calculated by utilizing (7.16), which allows us to determine the contribution of each source to the current outflow:

$$\hat{r}_{out}(k) = \sum_{i=1}^{n_s} \frac{\hat{v}_i(k)}{\hat{v}(k)} \cdot \hat{r}_i(k) \tag{7.27}$$

7.4 Case studies

7.4.1 Application on a transport network

In this case study, the “Hanoi” benchmark network [163], an example of a water transport network shown in Fig. 7.5, is used to demonstrate the chlorine concentration bounds generated by the BUBA. Realistic water demands were generated based on real observations of consumption at the level of a District Metered Area (DMA), and assigned at the network nodes. The system is simulated for a period of 24 hours, during which the demand at each node varies based on a unique pattern. The demands change every 5 minutes, which is equal to the discrete hydraulic-step used for this system.

Hydraulic model uncertainty is considered in the form of uncertain node demands and hydraulic model parameters. The uncertainty is considered to be bounded by a known bound. Specifically, for node demands the uncertainty upper bound is ±10% of the given value at each time step. Uncertainty in the hydraulic model is considered in the form of uncertain pipe roughness coefficients, with an upper bound of ±10% of the given value for each pipe.

Chlorine concentration, is regulated at reservoir node 1. A set-point chlorine booster regulates the concentration at that point to 0.4 mg/L. It is assumed that the initial concentration throughout the network is uniform and equal to 0.4 mg/L. The chlorine bulk reaction rate is constant in this example, and it is set to $r_i^b = -0.5$ 1/day. Wall reactions are set to $r_{ij}^w = -0.25$ m/day for all pipes l_{ij}. These are typical values from investigations on real systems when a first-order chlorine decay model is considered [100]. Relative diffusivity effects are ignored. Water quality is solved with a time step of $\Delta t = 1$ minute. Water-quality sensors in these systems are typically placed optimally, with respect to minimizing the impact of a potential contamination. Tools for sensor placement exist in the literature, an example of which is the *S-Place* toolkit [31]. The *S-Place* toolkit places the sensors by minimizing the impact

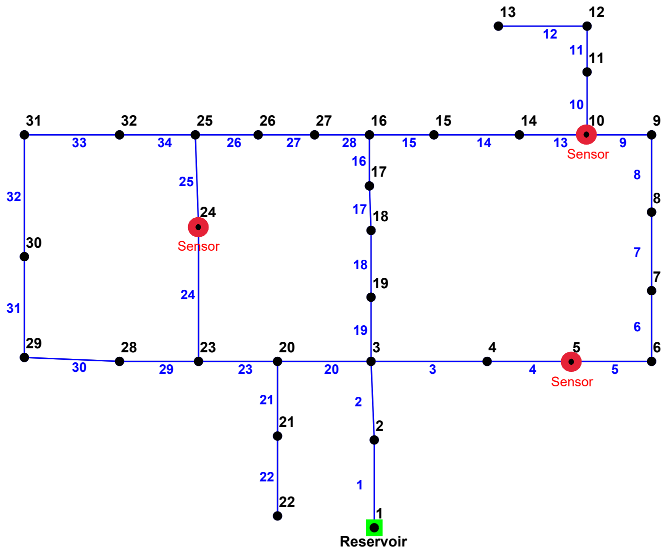


Figure 7.5. The "Hanoi" benchmark transport network and location of chlorine sensors.

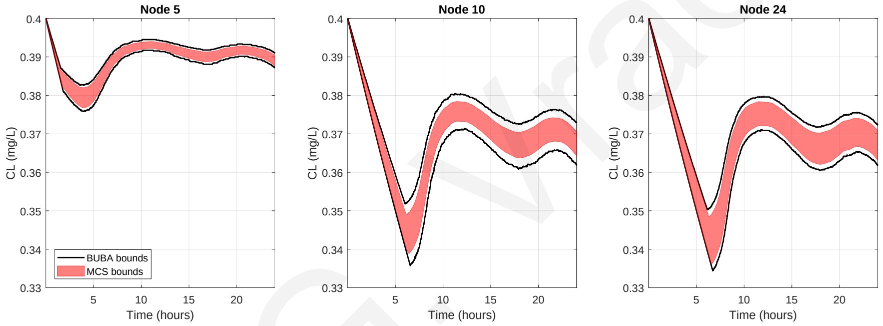


Figure 7.6. Chlorine concentration bounds generated by the proposed BUBA and 30000 Monte-Carlo simulations at three sensor locations on the "Hanoi" benchmark transport network.

of all possible contaminations, considering also modeling uncertainty. *Impact* in *S-Place* is defined as the volume of contaminated water consumed in m^3 until the contamination is detected. For the Hanoi network, three sensors were optimally placed using *S-Place* at nodes $\mathcal{N}_s = \{5, 10, 24\}$.

Water quality model uncertainty is considered on chlorine decay rates, i.e., bulk reaction rate r^b and wall reaction rate r^w. The uncertainty on these parameters is assumed bounded, with an upper bound of $\pm 10\%$ of their given value.

Chlorine concentration bounds were generated by the BUBA given the aforementioned case study parameters. To examine the validity of the bounds computed by the proposed BUBA, they are compared to bounds provided by extensive Monte-Carlo simulations (MCS) on the case study network, using the EPANET software [74] in MATLAB with the help of the EPANET-MATLAB toolkit [10]. In total 30000 simulations were performed for the Hanoi network, where the uncertain parameters were randomly varied inside their uncertainty bounds. In each simulation the minimum and maximum values of each quality state were saved. Quality tolerance in EPANET was set to 10^{-4}, to improve the accuracy of the calculated water quality state. The generated bounds on the locations of the three sensors are illustrated in Fig. 7.6. It is shown that BUBA provides tight bounds, close to those generated by the computationally intensive MCS.

7.4.2 Application on a District Metered Area network

In this case study, the "CY-DMA" network, shown in Fig. 7.7 is used to demonstrate the chlorine concentration bounds generated by the BUBA. "CY-DMA" is an example of a District Metered Area (DMA) network, as it was created based on a real DMA network in Cyprus. Realistic water demands were generated based on real observations of consumption at the level of a DMA, and assigned at the network nodes. For the "CY-DMA" network, nine sensors were optimally placed using *S-Place* toolkit at nodes $\mathcal{N}_s = \{3, 17, 28, 41, 44, 55, 65, 82, 90\}$. The rest of the case study parameters are the same as in the "Hanoi" network case study.

Chlorine concentration bounds were generated by the BUBA given the aforementioned case study parameters and compared to bounds generated by 30000 Monte-Carlo simulations. The results on the locations of the nine sensors are illustrated in Fig. 7.8. It is shown that the BUBA provides tight bounds on some states (nodes $3, 17, 44$), however in some states it provides wider bounds, such as at node 55 where the BUBA bound width is approximately five time larger than the MCS bound width. This is mainly due to the uncertainty in water flows, which can significantly affect chlorine concentration especially in the case when flow reversals occur. The BUBA bounds are generated considering all the possible cases explained by the system uncertainty, whereas it would take a large number of MCS to generate the extreme cases calculated by the BUBA. This illustrates the benefit of the proposed algorithm to *capture bounds that would be difficult to capture even with extensive simulations*.

7.4.3 Case study on real transport network

The proposed methodology is demonstrated using real data from a water transport network in a large city in Cyprus, shown in Fig. 7.9. This network has complete hydraulic state observability, achieved by flow measurements at the inflows and outflows of the network and a level sensor at the tank. It also contains three chlorine concentration sensors, one at the

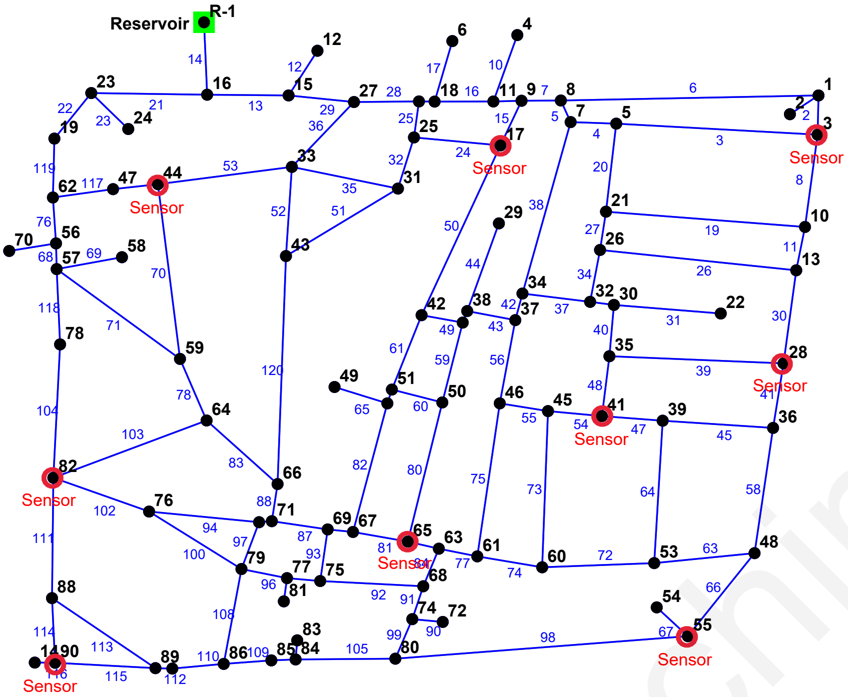


Figure 7.7. The “CY-DMA” district metered area network and location of chlorine sensors.

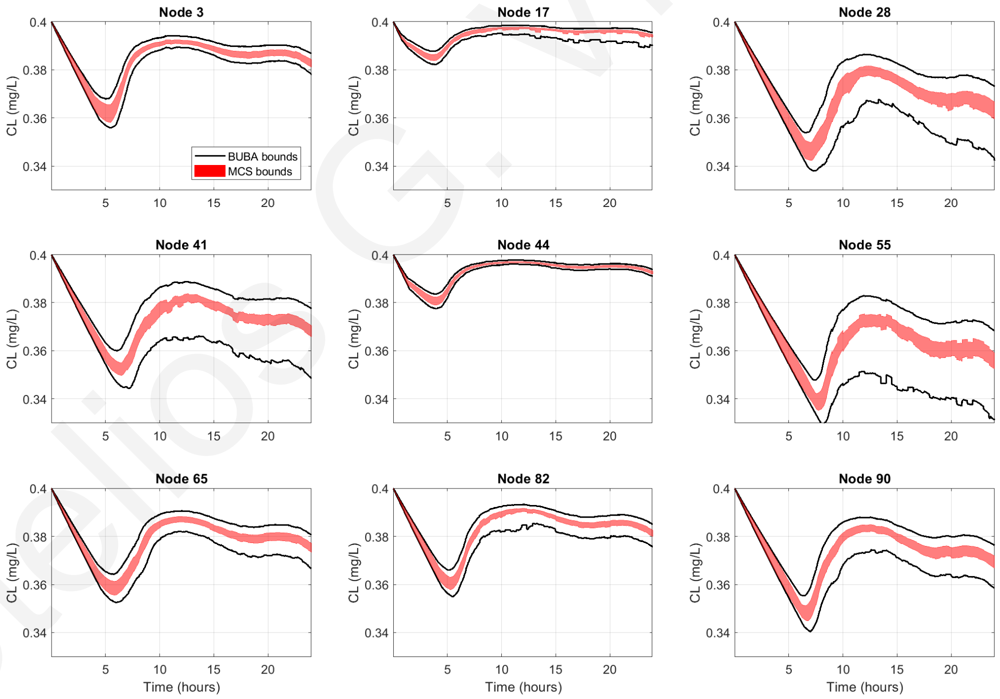


Figure 7.8. Chlorine concentration bounds generated by the proposed BUBA and 30000 Monte-Carlo simulations at nine sensor locations on the “CY-DMA” district metered area network.

supply tank inflow, one at the tank outflow and one at a network branch.

The tank volume observer described in section 7.3.2, uses the tank level measurements

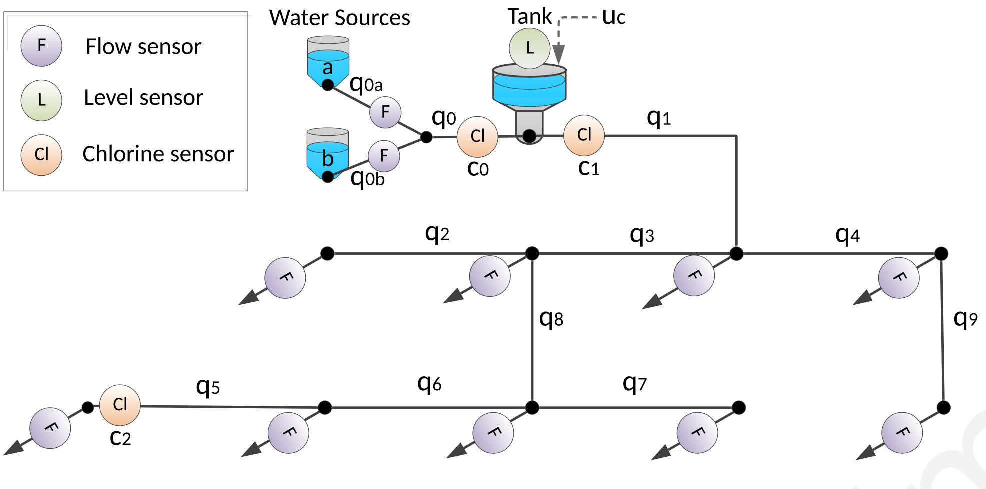


Figure 7.9. Diagram of the real transport network used in this case study

as well as the flow measurements $q_{0a}(k)$, $q_{0b}(k)$ and $q_1(k)$ to estimate the total water volume inside the tank $\hat{v}(k)$ at each time step, as well as the individual water volume from each of the two water sources, indicated by $\hat{v}_a(k)$ and $\hat{v}_b(k)$. The estimation of the two separate water volumes in the tank is shown in Fig. 7.10(c).

Chlorine sensors are installed at the inflow and outflow of the tank, with their measurements indicated by c_0 and c_1 respectively (Fig. 7.9). These measurements can be used to estimate the chlorine bulk reaction rate of water from each source, using the estimated water volumes, the chlorine measurements $c_0(k)$, $c_1(k)$ and the chlorine injection at the tank $u_c(k)$. First, an initial estimate of the reaction rates of each source is calculated using historical data. The optimization of (7.20) is solved, with conservative bounds on the decay rates set to $\overline{\boldsymbol{r}} = [0, 0]$ and $\underline{\boldsymbol{r}} = [-1, -1]$. The calculated reaction rates for each source are: $[\hat{r}_a(0), \hat{r}_b(0)] = [-0.0100, -0.0256]$ 1/hour. The chlorine concentration inside the tank can be estimated using these constant reaction rates; we will indicate the estimated chlorine concentration for this case by $\hat{c}_1^0(k)$. However, as seen from Fig. 7.10(a), there is significant discrepancy between the estimated $\hat{c}_1^0(k)$ and measured $c_1(k)$ concentration when using a constant reaction rate due to the varying water-quality characteristics in the tank.

The chlorine bulk reaction rate $r_{out}(k)$ at the tank output (and inside the tank) can be estimated in real-time using the parameter estimation algorithm described in Section 7.3. The parameter estimation algorithm estimates the reaction rates of each source, i.e. the reaction rate $\hat{r}_a(k)$ of source a and $\hat{r}_b(k)$ of source b, as seen in Fig. 7.10(b). These two reaction-rate estimates are then used with the water volume estimates $\hat{v}_a(k)$ and $\hat{v}_b(k)$ to calculate the total reaction rate $r_{out}(k)$ as described in (7.27). The estimated chlorine concentration in the

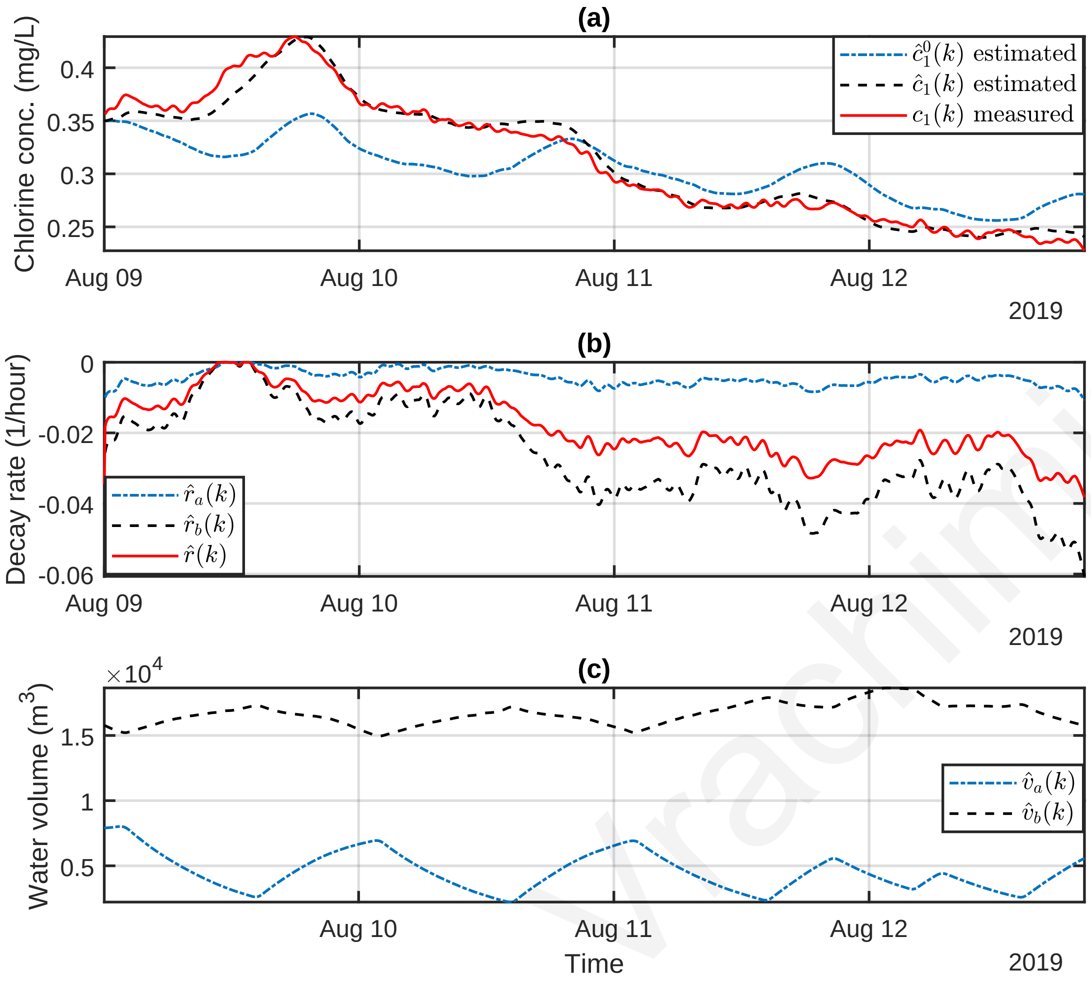


Figure 7.10. (a): Measurement of chlorine concentration at the tank output $c_1(k)$ compared to its estimation when using a constant bulk reaction rate $\hat{c}_1^0(k)$, and when using a time-varying bulk reaction rate $\hat{c}_1(k)$. (b): Estimation of chlorine reaction rates of source a and source b, and the estimated total reaction rate in the tank $\hat{r}(k)$. (c): Estimation of water volume from source a and source b in the tank.

tank $\hat{c}_1(k)$ using this time-varying reaction rate, is a better approximation of the measured concentration $c_1(k)$ than when a constant reaction rate is used, as illustrated in Fig. 7.10(a).

The proposed Backtracking Uncertainty Bounding Algorithm described in Section 7.2 is now used to calculate chlorine concentration bounds at the measured location c_2 inside the network, using the estimated time-varying tank bulk reaction rate $\hat{r}_{out}(k)$. The Iterative Hydraulic Interval State Estimator (IHISE) proposed in [2] is first used to calculate a bounded estimate of water flows in the network $\tilde{\boldsymbol{q}}(k) = [\underline{\boldsymbol{q}}(k), \overline{\boldsymbol{q}}(k)]$ using the available flow and head measurements. The IHISE considers 20% uncertainty on flow and tank level measurements. The BUBA uses the uncertain flows $\tilde{\boldsymbol{q}}(k)$, while it also considers 30% uncertainty on the given pipe wall reaction rates $\boldsymbol{r}^w$. The bulk reaction rate r^b of each water parcel tracked by

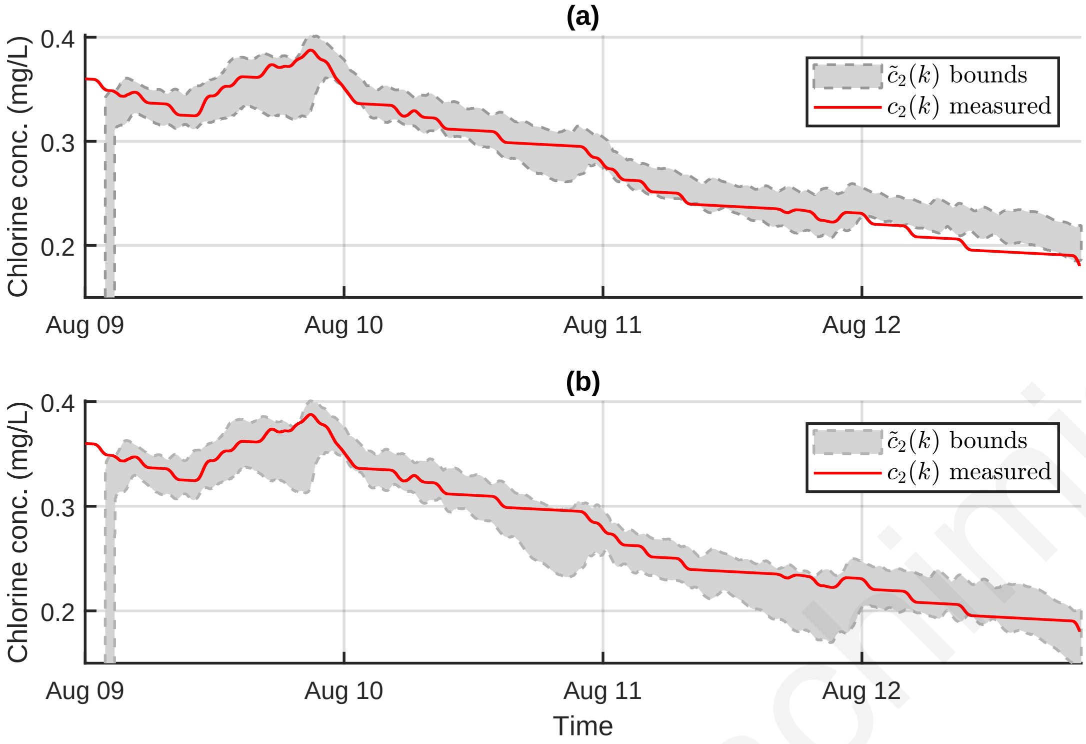


Figure 7.11. Measured chlorine concentration $c_2(k)$ compared to the BUBA calculated chlorine concentration bounds when (a) using a constant bulk reaction rate and (b) when using a time-varying bulk reaction rate which is estimated in real-time.

the BUBA, is equal to the tank bulk reaction rate which is given by the parameter estimation algorithm, i.e., for a water parcel i which is exiting the tank at time step k then $r_i^b = \hat{r}_{out}(k)$. Moreover, 30% uncertainty is considered on the bulk reaction rate r_i^b of each water parcel.

The chlorine concentration bounds $\tilde{c}_2(k)$ calculated by the BUBA are compared in Fig. 7.11 to the measured concentration $c_2(k)$ in the network . In the same figure, the benefit of estimating in real-time the reaction rate in the supply tank is examined by comparing the BUBA bounds when using a constant reaction rate in the tank (Fig. 7.11(a)), with the BUBA bounds when using the estimated time-varying reaction rate (Fig. 7.11(a)). The results show a slight improvement on the bounding estimates when using the estimated time-varying decay rate, as they better enclose the measurements, which are assumed to be fault-free during the examined period. Note that for the specific case study, the bounds generated by the BUBA are equal to zero for the first 2 hours and 45 minutes. This is due to the travel time of chlorine from the tank to the measured branch of the network. Before this time, the BUBA gives zero bounds because of the assumption that the initial chlorine concentration in the network was zero.

7.5 Conclusions and future work

This work presents a comprehensive methodology for generating bounded chlorine concentration estimates at sensor locations in WDN, suitable to be used for sensor fault and contamination detection purposes. The proposed Backtracking Uncertainty Bounding Algorithm (BUBA) is able to calculate tight chlorine concentration bounds, given bounded hydraulic-state estimates and bounds on water quality parameters. The bounds are validated and compared to a large number of Monte-Carlo simulations on two benchmark networks. Additionally, the BUBA is designed to facilitate time-varying bulk reaction rates, which occur in systems with multiple water sources. This is demonstrated by designing a real-time bulk reaction rate estimation algorithm in tanks with water originating from different sources. The parameter estimation algorithm and the proposed BUBA are then applied on a real network case study with real data. An improvement is shown in the estimated chlorine concentration bounds when using the real-time bulk reaction rate estimates.

The proposed methodology for bounded chlorine concentration estimation uses a linear, first order model for chlorine decay dynamics. In future work we will demonstrate the formulation of this methodology when using a more complex model of chlorine decay. Moreover, we have only demonstrated a real-time parameter estimation algorithm for the decay rates in tanks, however the proposed methodology is able to use the path information provided by the BUBA to also facilitate a wall reaction coefficient estimation algorithm, which will further reduce parameter uncertainty. The quantification of the parameter estimation uncertainty in order to properly define parameter uncertainty bounds during the parameter estimation process should also be investigated.

The execution time of the BUBA depends on the length of the water flow paths contributing chlorine to each examined node. The flow path lengths depend in turn on the size of the examined network, the distance of known concentration points from the examined nodes, and the travel time of water parcels with the worst case scenario being the use of the initial network conditions. The execution times can be improved significantly by utilizing more memory resources to save the calculated chlorine concentrations along the node paths at each corresponding time-step and use this information in subsequent time-steps, thus avoiding the need to recalculate them.

Chapter 8

Contamination detection using chlorine concentration thresholds

To monitor water quality, utilities typically employ periodic manual sampling. However, when a contamination event occurs, it may require days before it is detected. To enhance monitoring, utilities employ sensors which monitor various water quality parameters. A common approach is the use of chlorine sensors for monitoring chlorine residuals at different locations in the network, in order to determine whether a contamination event has occurred. Unfortunately, due to significant variability in water demands, as well as the effect of hydraulic and quality control actions, the disinfectant residual at the sensor location may fluctuate significantly in time, and therefore, model-free event detection algorithms may not be able to detect certain contamination events, or they may cause false alarms. In [126], a method for computing bounds of the expected chlorine concentration at different chlorine sensing locations is presented. This method considers the known chlorine input injection signals and also takes into account the uncertainties in hydraulic dynamics. The bounds are calculated using multiple Monte-Carlo Simulations (MCS) which run in parallel to the real system.

The present work extends the work in [126] by considering the use of multi-level thresholds, which facilitate the computation of warnings regarding the possible occurrence of a contamination event. Moreover, it is shown how these multi-level thresholds can be created without the use of MCS, by utilizing the analytical methodology of the BUBA presented in Chapter 7. At each time-step, a low-concentration bound is estimated by considering uncertainty on water demands and roughness coefficients. Based on this bound, multiple levels (thresholds) are calculated in real-time, giving the ability to identify possible contamination events in less time than the baseline Lower-bound method, while contributing to the overall risk-evaluation process, limiting false positives. The proposed algorithm is evaluated using a randomized contamination event generator implemented on EPANET[74] and EPANET-MSX[98], based on a benchmark network.

This chapter is organized as follows: In Section 2, the design methodology is described, and in Section 3, a case study on a realistic water distribution system is presented. In Section 4 we present how MCS can be replaced by the BUBA methodology of Chapter 7. Finally, Section 5 concludes this work and discusses future work.

8.1 Design Methodology

The Water Distribution Network has actuators and sensors for controlling and monitoring chlorine concentrations, and is driven by the partially-known consumer demands $\boldsymbol{d}(k) \in \mathbb{R}^{n_n}$ at n_n nodes. The output signal $\boldsymbol{y}_c(k) \in \mathbb{R}^{n_s}$ of the system corresponds to the measured chlorine concentrations at n_s locations where chlorine concentration sensors are installed.

We define k as the discrete time, with sampling time Δt, $\boldsymbol{x}(k)$ as the average chlorine concentration state vector and $\boldsymbol{w}(k)$ as the average contaminant concentration state vector, where each state corresponds to the concentration in a finite volume within a pipe. Furthermore, we define $\boldsymbol{u}_c(k)$ as the controllable disinfectant concentration input and $\boldsymbol{d}(k)$ the consumer water demands. Following on the formulation in [126], the water distribution system water quality model can be described by

$$\begin{aligned} \boldsymbol{x}(k+1) &= f_c(\boldsymbol{x}(k), \boldsymbol{u}_c(k), \boldsymbol{d}(k); \boldsymbol{\theta}) + g_c(\boldsymbol{x}(k), \boldsymbol{w}(k); \boldsymbol{\theta}) \\ \boldsymbol{w}(k+1) &= f_w(\boldsymbol{w}(k), \boldsymbol{d}(k); \boldsymbol{\theta}) + g_w(\boldsymbol{x}(k), \boldsymbol{w}(k); \boldsymbol{\theta}) + \phi(k; \boldsymbol{\theta}_\phi) \\ \boldsymbol{y}(k) &= C\boldsymbol{x}(k), \end{aligned} \tag{8.1}$$

where $f_c(\cdot)$ and $f_w(\cdot)$ are the advection functions for the disinfectant and contaminant substances respectively; $g_c(\cdot)$ and $g_w(\cdot)$ are the reaction functions for the disinfectant and contaminant substances respectively. The vector $\boldsymbol{\theta}$ is in general comprised of all the parameters affecting the dynamics, and in this work we will consider pipe roughness coefficients; in general, the actual parameters of the vector $\boldsymbol{\theta}$ are unknown (or partially/nominally known) and may be time-varying. Function $\phi(\cdot)$ corresponds to the unknown non-negative contamination fault function; the set $\boldsymbol{\theta}_\phi$ is comprised of the contamination fault parameters, such as starting time, duration and magnitude. Finally, $\boldsymbol{y}_c(k)$ is the output vector of the chlorine measurements and C is the output matrix. Note that, this mathematical representation of the system is equivalent to the Eulerian model [91, 164] and the differential equations used by EPANET-MSX [98].

Typically, the Water Distribution Network has a controller responsible for computing the input signals $\boldsymbol{u}_c(k) \in \mathbb{R}^{n_u}$ for the n_u chlorination actuators. In this work, without loss of generality, a time-based control algorithm is considered, to specify the chlorine concentration set-point at the disinfection locations. The input signal $\boldsymbol{u}_c(k)$, which can be time-varying, is known and will be considered in the event detection algorithm. The input signal is a key parameter of the proposed methodology, as it allows to consider known variations in the quality, without triggering false positives alarms.

The actual nodal water demands $\boldsymbol{d}(k) \in \mathbb{R}^{n_n}$ at the n_n nodes which affect the contaminant and disinfectant propagation, are not known, but in general are bounded within a region $\boldsymbol{d}(k) \in \tilde{\boldsymbol{d}}(k) \triangleq \left[\underline{\boldsymbol{d}}(k), \overline{\boldsymbol{d}}(k)\right]$. Likewise, the system parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ at the n_p pipes, are partially known and are bounded within a region $\boldsymbol{\theta} \in \tilde{\boldsymbol{\theta}} \triangleq \left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$. We assume that it is possible to calculate lower and upper bounds for the regions $\tilde{\boldsymbol{d}}(k)$ and $\tilde{\boldsymbol{\theta}}$. Furthermore, the output vector is bounded within a region, $\boldsymbol{y}(k) \in \tilde{\boldsymbol{y}}(k) \triangleq \left[\underline{\boldsymbol{y}}(k), \overline{\boldsymbol{y}}(k)\right]$, which is unknown, however an estimate can be calculated using the complex and uncertain dynamics of hydraulics and quality.

For the event detection, it is useful to calculate an estimate of the lower bound of the output vector, such that $\boldsymbol{y}(k) \geq \underline{\boldsymbol{y}}(k)$. This can be achieved using randomized Monte-Carlo simulations [126]; in specific, M randomized simulations can run in parallel to the operation of the real system, based on a nominal water distribution system model, and a nominal water demand model while taking into account the input signal $\boldsymbol{u}_c(k)$. Note that at each time step, only the next time step needs to be computed. In specific, for the i-th Monte Carlo simulation, vectors $\boldsymbol{d}^{(i)}(k)$ and $\boldsymbol{\theta}^{(i)}$ are calculated, such that $\boldsymbol{d}^{(i)}(k) \in \tilde{\boldsymbol{d}}(k)$ and $\boldsymbol{\theta}^{(i)} \in \tilde{\boldsymbol{\theta}}$. Therefore, the i-th Monte Carlo simulation is used to calculate an estimate of the concentration at the j-th sensor, $\hat{y}_j^{(i)}(k)$ at time k using (8.1). Finally, the estimated lower-bound can be calculated based on the M simulations, as

$$\underline{y}_j(k) = \min\left\{\hat{y}_j^{(i)}(k; \boldsymbol{d}^{(i)}(k), \boldsymbol{\theta}^{(i)}) \ : \ i \in \{1, \ldots, M\}\right\}. \tag{8.2}$$

In practice, due to the uncertainties and the complex nature of the system, it cannot be guaranteed that $\underline{y}_j(k) < y_j(k)$ for all k. Therefore, it is not possible to construct a detection scheme which is based on the violation of the lower-estimation threshold, as it would cause a large number of false positives. A different approach, would be to consider multiple lower thresholds, and to require a delay threshold in detecting an event. The intuition is that lower

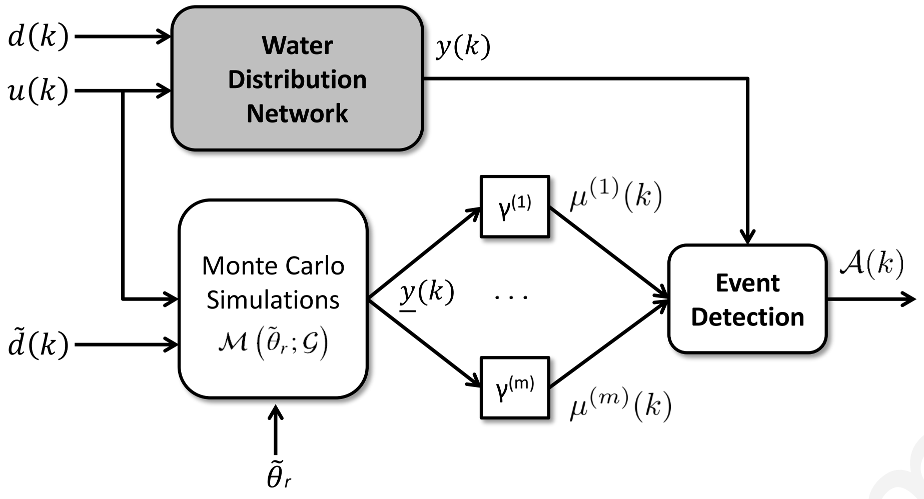


Figure 8.1. Architecture of the proposed contamination event detection scheme.

levels would be able to detect a smaller number of (larger) contamination events, but faster, whereas higher levels would be able to detect a larger number of (smaller) contamination events but with some delay as well as with a higher risk of false positive alarms.

In this work a multi-level approach is proposed as follows: Compute multiple detection levels $\mu^{(i)}(k)$, for $i \in \{1,\dots,m\}$, such that $\mu^{(i)}(k) = \gamma^{(i)}\underline{y}_j(k)$ and for $0 < \gamma^{(1)} < \gamma^{(2)} < \dots < \gamma^{(m)} \leq 1$. Then, for each sensor j and for each level i, compute a detection delay threshold $h_j^{(i)}$ based on a set of historical and simulation data $\mathcal{S}$ describing both the normal and faulty behavior of the system, for optimizing a specific metric (e.g., minimizing number of False Positives or maximizing Accuracy and F1-score); such that $h_j^{(i)} = f_h(\mathcal{S})$, where f_h is the algorithm which selects the optimal delay threshold.

The event detection logic compares a window of measurements with the multiple levels. In general, an event alarm flag $\mathcal{A}_j(k)$ at the j-th sensor node is triggered at time k, if for a time-window of measured chlorine concentrations at the j-th sensor node, there is a number of consecutive measurements which are lower than a certain level. In specific,

$$\mathcal{A}_j(k) = \begin{cases} Warning \quad y_j(\tau) < \mu_j^{(i)}(\tau),\ \tau \in \{k - h_j^{(i)},\dots,k\},\ i \in \{1,\dots,m\} \\ Normal \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{8.3}$$

The overall architecture described in the paragraphs above can be summarized in Fig. 8.1.

8.2 Case Study

In the simulation studies of this section, a hydraulic benchmark model based on EPANET and a quality benchmark model based on EPANET-MSX are used, similar to the case study

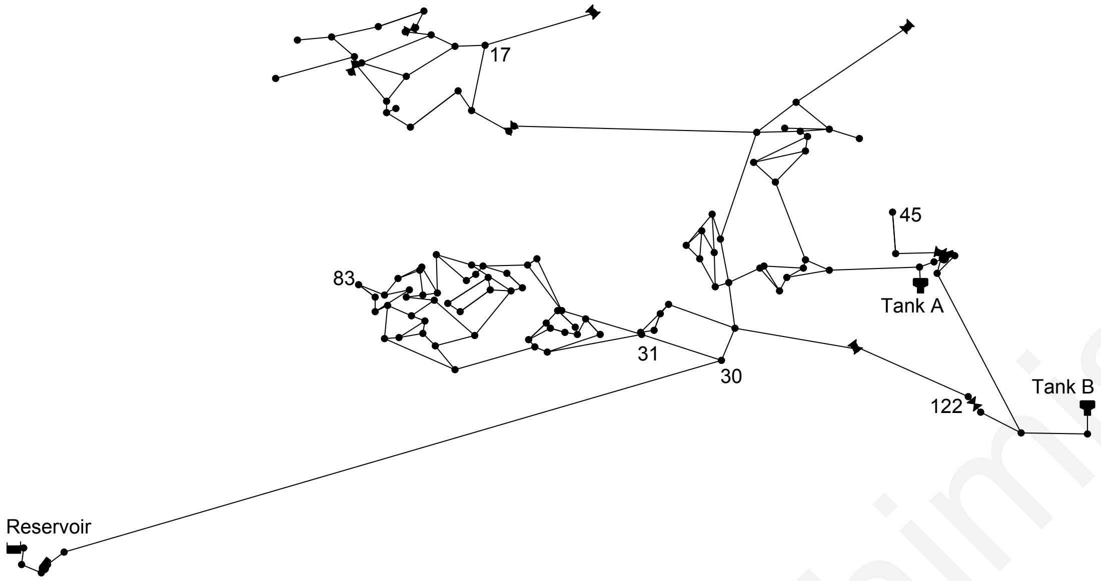


Figure 8.2. The benchmark water distribution system with 129 nodes.

presented in [126]. The hydraulic benchmark corresponds to the ‘Network 1’ which was used as part of the “Battle of the Water Sensor Networks” design competition [108], depicted in Fig. 8.2. The network is comprised of $n_p = 178$ pipes, $n_n = 126$ junctions, two tanks and one reservoir. Realistic parameters are considered for all structural and hydraulic characteristics. For each junction, the benchmark assigns a nominal water consumption volume, $\boldsymbol{d}(k)$, of 48-hours duration and with 30-minute hydraulic time-step. The quality benchmark is based on the model used in [123] for simulating reactions of Chlorine and Arsenite in drinking water. The water distribution network is assumed to use chlorine disinfection, and chlorine disinfection actuators are considered at two locations, $n_u = 2$, the ‘Reservoir’ and ‘Tank A’, as shown in Fig. 8.2. In this case-study, all the hydraulic and quality models are simulated using the EPANET and the EPANET-MSX toolkits, within the MATLAB programming environment using the EPANET-MATLAB Toolkit, as described in Appendix A'. The multi-species solver uses the 5th-order Runge-Kutta method with a 10-minute time-step.

The placement of the five chlorine sensors is based on a Pareto optimal solution proposed in the “Battle of the Water Sensor Networks” competition [108]. In specific, $n_s = 5$ water quality sensors are considered at nodes ‘17’, ‘31’, ‘45’, ‘83’ and ‘122’, as indicated in Fig. 8.2. Considering single contamination events which can occur at any node at any time within one day, with 5-minute sampling time, in total 37,152 contamination scenarios can be constructed (129 nodes $\times$ 288 samples), for which the sensor placement scheme used is able to detect (given an ideal contaminant sensor), 75.6% of all the constructed scenarios. This means that

24.4% of all constructed contamination events cannot be detected using this sensor placement scheme, and thus more sensors would be required to increase coverage.

The case study involves the simulation of 8 days of operation of the realistic water distribution network. In the following, it is assumed that the demands and roughness coefficients are unknown, but are bounded within a known region. Furthermore, the controller is based on a simple known scheduling rule for specifying the input signal, e.g., due to a change in water source quality, as follows:

$$\boldsymbol{u}_c(k) = \begin{cases} 1 \text{ mg/L} & \text{if } k < 3 \text{ days}, k \geq 5 \text{ days} \\ 0.5 \text{ mg/L} & \text{if } 3 \text{ days} \leq k < 5 \text{ days} \end{cases} \tag{8.4}$$

The evaluation methodology followed in this case study is described as follows:

1. Identify the 10% worst-case contamination events for the network (with respect to the impact metric of contaminated water consumption volume). Construct a contamination event scenarios set, comprised of the location and the time of occurrence.
2. Construct a dataset of 500 contamination scenarios based on the contamination event scenarios set, with randomized magnitudes $0 < \phi(\cdot) \leq 1$ occurring on Day 4 of the simulation, and with 15% maximum uncertainty in the demands (patterns and base demands) at each node, as well as the roughness coefficients.
3. Construct a dataset of 500 scenarios of normal operation without contamination events and with randomized demands and roughness coefficients, as in Step 2. This is important in order to evaluate the event detection ability of the proposed algorithm.
4. At each time step, run $M = 100$ Monte Carlo simulations to estimate the Chlorine concentration at the next time-step. For this, we consider $\tilde{\boldsymbol{d}}(k) = (1 \pm 20\%)\,\hat{\boldsymbol{d}}(k)$ uncertainty in the demands and $\tilde{\boldsymbol{\theta}} = (1 \pm 20\%)\,\hat{\boldsymbol{\theta}}$ in the roughness coefficients, with respect to the nominal estimated parameters $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{\theta}}$. From the set of the 100 concentration estimations at each node, calculate the minimum lower-bound concentration estimation.
5. Based on the minimum lower-bound concentration estimation, compute m multi-level thresholds which will be used for detection at each time-step.
6. To measure the ability of the detection scheme under different parameters, the following are considered: For each scenario, a label is assigned depending on whether the event has been correctly identified (True Positive), if there was a false alarm (False Positive), if it was missed (False Negative), or if there was no contamination affecting the sensor and there was no alarm (True Negative). This is performed for each sensor

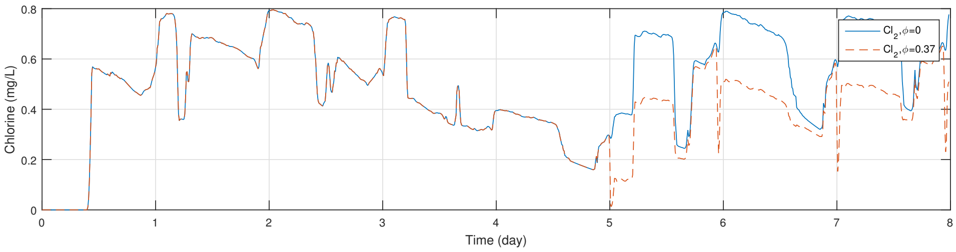


Figure 8.3. Comparison of chlorine concentration in sensor at node '17' during normal operation and during a contamination event.

separately. It is important to note that in some contamination events, the contaminant may not reach some sensor locations. In addition to the labeling of each node behavior for each scenario, we consider the general case when all the information from the sensors is combined. Specifically, if there is at least one sensor with 'False Positive', then the detection scheme as a whole is labeled as 'False Positive'; otherwise, if there is at least one 'True Positive', then the detection scheme is 'True Positive'; otherwise, if there is at least one 'False Negative', it is classified as 'False Negative'; otherwise, it is 'True Negative'. This approach penalizes false positive warnings.

7. Based on the 1000 simulated training scenarios with and without contamination events, run the detection scheme for various levels and detection delay thresholds, and evaluate their True Positive, True Negative, False Positive and False Negative score, as well as their F1-score metric.
8. Calculate the minimum detection delay thresholds which minimize the number of False Positives (i.e., zero False Positives) for each level i, $\gamma^{(i)} \in \{60\%, 62\%, ..., 100\%\}$, $i \in \{1, ..., m\}$ and $m = 21$.
9. Finally, evaluate the effectiveness of the proposed detection scheme based on the calculated thresholds, in a test set of 357 contamination events, with unknown hydraulic and water quality conditions, which were not used during the training period.

An example of how chlorine concentration is affected due to contamination event, is shown in Fig. 8.3, when Arsenite is injected at node '27' on Day 4 at time 08:30 as a step input of magnitude 0.37 mg/L. At Day 5, the effects of the Arsenic contamination are appearing on the sensor measurement, resulting in a decrease of the normal chlorine concentration. It is important to note that due to the variability in the chlorine input, the normal measured output is between 0.2-0.8 mg/L, and the contamination event causes a violation of these bounds for only a short period of time. Thus, it might not be possible to confirm the event simply based on the use of fixed alarm thresholds. Figure 8.4 depicts the estimated chlorine concentrations

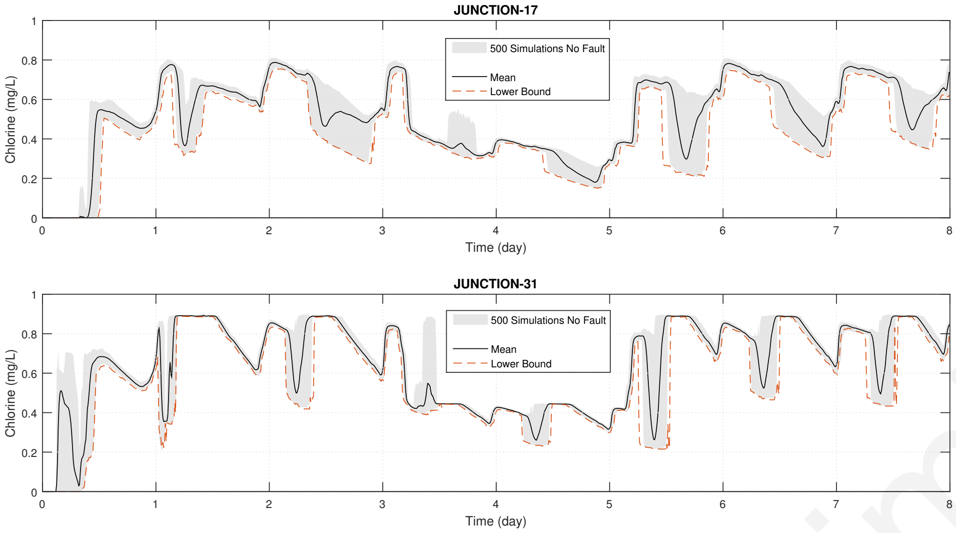


Figure 8.4. Comparison of 500 randomly simulated chlorine concentration estimations at two sensor locations and the estimated lower bound computed using the 100 Monte Carlo simulations

measured at nodes '17' and '31', for 500 randomized simulations (depicted as grey area). The lower bound is depicted with the dashed line, and is the result of the computation of 100 Monte Carlo simulations which run in parallel to the operation of the system, considering the inputs to the system. It is important to note that the lower bound is an estimate and as a result, it cannot be assumed that it will not be violated at certain time steps. Figure 8.5 depicts the estimated detection delay threshold computed based on the 1000 simulation scenarios, which minimize the number of False Positives (in this case study we require zero False Positives), with respect to the percentage level of the Lower Bound computed using the 100 Monte Carlo simulations. Figure 8.6 depicts a subset of the multiple levels computed for node '17', together with the measured signal when a contamination event has occurred and reached the node. As seen from the graph, the event is detected at Day 4.48, with 17 time steps delay (i.e., 170 minutes).

Table 8.1 illustrates the results of the testing set of 357 random scenarios at the 10% worst-case locations and times for contamination, when considering the multi-level detection thresholds. The first part of the table illustrates the results for each sensor separately, with respect to the True/False Positives/Negatives. When considered independently, node '17' has the highest F1-score (harmonic mean between precision and recall), and node '83' the worst F1-score. From the 357 scenarios constructed, there were 2 False Positive occurrences at two nodes. However, when the results are combined according to the rule specified

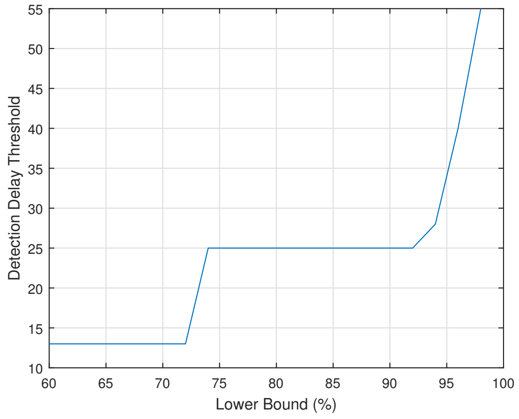


Figure 8.5. Detection delay thresholds with respect to the percentage level of the Lower Bound.

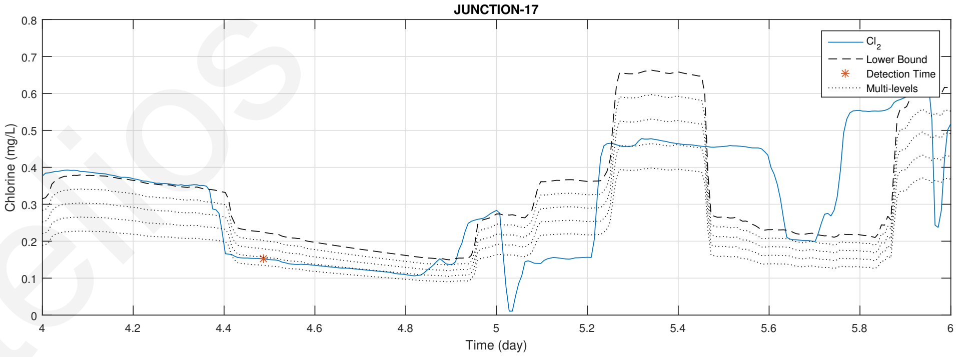


Figure 8.6. The chlorine concentration at node '17' affected by a contamination fault, with respect to the lower bound estimated using the Monte Carlo simulations and the Multiple-bounds based on the proposed methodology. The star sign indicates the time the algorithm has detected the contamination event.

in the methodology, which penalizes False Positives, more scenarios are classified as True Positives than what each sensor location achieves independently. This is because some contamination scenarios can be detected only by some sensors. As a result, a higher F1-score can be achieved when combined the 5 sensor information. As a note, the 34 True Negatives in the 'combined' case corresponds to the cases when the contamination event did not arrive at the sensor locations at a concentration greater than zero within the time period of the simulation.

	True Positive	False Positive	False Negative	True Negative	F1-Score
Junction-17	232	0	85	40	0.85
Junction-31	213	0	102	42	0.81
Junction-45	178	1	137	41	0.72
Junction-83	154	0	162	41	0.66
Junction-122	211	1	108	37	0.79
Combined	250	2	71	34	0.87

Table 8.1. Event detection algorithm evaluation on 357 random scenarios.

Table 8.2 shows the detection delay results when using the Lower-bound method and the Multi-level method. To compare both methods, the minimum detection delay threshold was selected for the Lower-bound method, so that the same number of False Positives (i.e., 2) are found when using the test set. The results demonstrate that the Multi-level method is able to detect contamination events significantly faster than if only the Lower-bound method was used. In general, for the Multi-level method, detection occurs with a delay of median value of 2.33 hours, and a maximum delay of 46.33 hours. It is important to note that these delays are required in order to guarantee that the False Positives are minimized, as it causes the method to use conservative thresholds. An interesting observation of the present case study is that the contamination magnitudes of the detected scenarios compared with the undetected scenarios are not significantly different with respect to their medians, when using the Wilcoxon rank sum test (p=0.448).

Method	Minimum	Median	Average	Maximum
Lower Bound	18.33	18.66	26.00	57.50
Multi-level	2.17	2.33	5.43	46.33

Table 8.2. Detection delay for the Lower-bound method and the Multi-level method (hours).

8.3 Contamination detection using bounded water-quality state estimation

The methodology of creating chlorine concentration thresholds using Monte-Carlo simulations has some disadvantages: First, the thresholds created by MCS are not guaranteed to include all the possible outputs of the system, as defined by the bounds on the uncertain parameters, as that would require a very large number of simulations. The application of the contamination detection algorithm using MCS in real-time is limited by the computing capabilities of the platform hosting the algorithm. Second, even though the real-time input of the system can be used in the simulations, it is not straightforward to use real-time measurements from the system which can improve the calculated thresholds. This would require a state-estimation algorithm which is able not only to use measurements to estimate the water-quality state, but also create thresholds on state estimates.

Such a state estimation algorithm is the bounded chlorine concentration estimator as proposed in Chapter 7. The proposed BUBA methodology uses real-time hydraulic interval-state estimates produced by the IHISE algorithm as described in Chapter 4. These interval estimates consider uncertainty in water demands and pipe parameters such as roughness coefficients, and it is able to use hydraulic-sensor measurements to improve these estimates. The BUBA then uses bounded water flow estimates, while considering uncertainty on water-quality modeling in the form of uncertain decay rates, to calculate thresholds on chlorine concentration. Moreover, the BUBA is able to utilize chlorine concentration measurements to improve the provided water-quality model using real-time learning of decay rates. The advantages using the IHISE and BUBA methodologies compared to MCS are summarized as follows:

- The use of bounded water flow estimates to analytically calculate bounds on chlorine concentration which are guaranteed to include the actual concentration in the contamination-free case given the provided water-quality model and uncertainty on decay rates

- The utilization of real-time hydraulic measurements to improve bounds on the uncertain water flows
- The use of real-time parameter estimation to improve the estimates of water-quality parameters such as chlorine decay rates.

Multi-level thresholds can be created using the analytical methodologies of IHISE and BUBA. Here, we calculate different thresholds by changing the level of uncertainty of the input parameters. This is an advantage of having an analytical method of calculating thresholds compared to MCS, as for each level of input and parameter uncertainty we can calculate in one execution the corresponding output thresholds. Specifically, the detection level $\boldsymbol{\mu}^{(i)}(k)$, for $i \in \{1, \ldots, m\}$ is calculated as follows:

$$\boldsymbol{\mu}^{(i)}(k) = \tilde{f}_c\Big(\boldsymbol{c}(k-1), \boldsymbol{u}_c(k), \tilde{\boldsymbol{q}}^{(i)}(k); \mathcal{M}(\tilde{\boldsymbol{\theta}}_r^{(i)}(k); \mathcal{G})\Big), \tag{8.5}$$

where $\tilde{f}_c(\cdot)$ is the bounding chlorine concentration estimator as defined in (7.4), $\tilde{\boldsymbol{q}}^{(i)}$ are the flow bounds for threshold i, and $\tilde{\boldsymbol{\theta}}_r^{(i)}(k)$ are the water-quality parameter bounds for threshold i. The flow bounds are calculated using the IHISE methodology of Chapter 4 as follows:

$$\tilde{\boldsymbol{q}}^{(i)}(k) = \tilde{f}_h\Big(\boldsymbol{y}_h(k), \boldsymbol{u}_h(k), \tilde{\boldsymbol{d}}^{(i)}(k); \mathcal{M}(\tilde{\boldsymbol{\theta}}_h^{(i)}; \mathcal{G})\Big), \tag{8.6}$$

where $\boldsymbol{y}_h(k)$ are the hydraulic sensor measurements, $\tilde{\boldsymbol{d}}^{(i)}(k)$ are the water demand bounds, and $\tilde{\boldsymbol{\theta}_h}^{(i)}(k)$ are the hydraulic parameter bounds. We define $\gamma^{(i)}$ as the level of uncertainty such that $0 < \gamma^{(1)} < \gamma^{(2)} < \ldots < \gamma^{(m)} \leq 1$. Given estimated values for demands $\hat{\boldsymbol{d}}(k)$, hydraulic parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{h}}(k)$ and water-quality parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{r}}(k)$, the corresponding input uncertainty for threshold $\mu^{(i)}(k)$ is given by: $\tilde{\boldsymbol{d}}^{(i)}(k) = \pm\gamma^{(i)}\hat{\boldsymbol{d}}(k)$, $\tilde{\boldsymbol{\theta}}_{\boldsymbol{h}}{}^{(i)}(k) = \pm\gamma^{(i)}\hat{\boldsymbol{\theta}}_{\boldsymbol{h}}(k)$ and $\tilde{\boldsymbol{\theta}}_{\boldsymbol{r}}{}^{(i)}(k) = \pm\gamma^{(i)}\hat{\boldsymbol{\theta}}_{\boldsymbol{r}}(k)$.

The overall architecture describing the multi-level contamination detection methodology using the IHISE and BUBA state estimators is summarized in Fig. 8.7.

8.4 Conclusions and Future Work

In this work we present a model-based contamination event detection algorithm using multiple detection thresholds, which monitor multiple chlorine concentration measurements in real-time. In parallel to the real system operation, a set of Monte Carlo simulations are executed, while at each time-step, based on the known input and by assuming known bounds

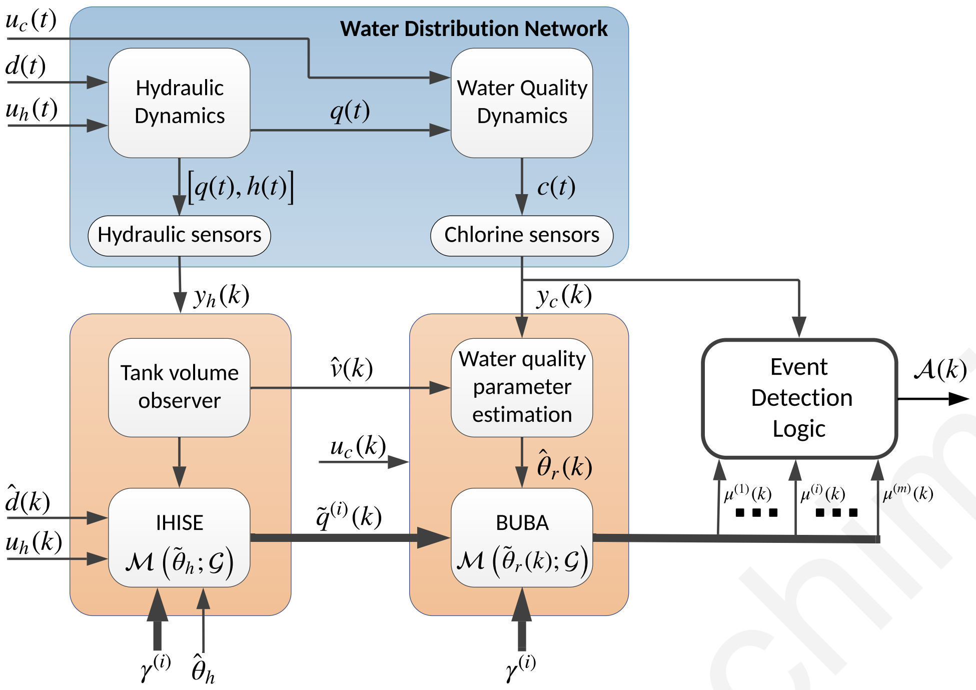


Figure 8.7. Contamination detection using the multi-level thresholds and IHISE and BUBA state estimators.

on the demands and roughness coefficients, an estimate of the chlorine concentration lower-bound is calculated. However, this lower-bound cannot guarantee that the chlorine measurements will be larger at every time step, due to the complex uncertain dynamics. Instead, a detection delay threshold can be estimated, based on historical or simulated data. However, when considering the lower-bound, this threshold can be large in order to reduce false positives. To improve this, a multi-level threshold approach is adopted in this work. In specific, a set of pre-defined levels are computed as percentages of the lower-bound estimate, and for each level a detection delay threshold is calculated based on a training dataset describing normal and faulty conditions.

To evaluate the results, a training set of 1000 scenarios was created, with 500 normal and 500 contamination scenarios, and based on these, the detection delay thresholds were computed for each level. Afterwards, the algorithm was evaluated using a test set of 357 random contamination scenarios. The results, at least for this case study, demonstrate the ability of the algorithm to identify a number of contamination events with a very small number of false positives and with significant reduction in the detection delay.

Moreover, we show how to generate multiple level thresholds using the analytical BUBA methodology in combination with hydraulic interval-state estimation, which in contrast to

MCS it is able to guarantee the inclusion of the actual concentration in the contamination-free case given the provided water-quality model. Additionally, this approach is able to utilize available measurements from the network to improve the generated thresholds.

Future work will investigate the application of the algorithm in a water quality monitoring system, and evaluate its effectiveness in real operational conditions.

Chapter 9

Active contamination detection

In this paper, we propose a novel methodology for altering the area monitored by water quality sensors in Water Distribution Systems (WDS) when there is suspicion of a contamination event. The proposed Active Contamination Detection (ACD) scheme manipulates WDS actuators, i.e., by closing and opening valves or by changing the set-points at pressure controlled locations, to drive flows from specific parts of the network in predetermined paths, and enable the sensors to monitor the quality of water from previously unobserved locations. As a consequence, the monitoring coverage of the sensors is increased and some contamination events occurring within those areas can be detected. The objective is to minimize the contamination impact by detecting the contaminant as soon as possible, while also maintaining the hydraulic requirements of the system. Moreover, the methodology facilitates the isolation of the contamination propagation path and its possible source. We demonstrate the ACD scheme on two networks, analyze the results and open the discussion for further work in this area.

9.1 Introduction

Traditionally, one of the most popular approaches for contamination detection in WDS uses a set of fixed sensor locations over the WDS to monitor and alert faults [165, 166]. This type of event detection methodology is also common in other infrastructure systems such as power, transportation and communication networks. For each given WDS topology, a set of possible sensor locations is available and the problem of finding the optimal set has been investigated in depth over the past decades [88, 167, 168, 109]. The methodology of fixed sensor locations for fault detection can be improved, in terms of detectability, by choosing optimal sensor locations based on hydraulic and topological analysis of the system. Sensor placement for monitoring faults takes into consideration, mostly, four main goals: (1) maximal coverage of system components, (2) early detection of fault events, (3) deriving information on the event source and (4) minimal number of sensors for economic reasons. Since the resulted set of sensor locations applies minimal to no change on the network topology or hydraulics, it is regarded as a Passive Contamination Detection (PCD) scheme, in which no manipulation on the systems' original (pre-setup) operation condition is being made. The topology and hydraulics of the WDS are regarded as given and cannot be changed or altered for detectability purposes.

In search for improved detection methodologies, studies in which the locations of the above sensors are perfected to better achieve the four main goals of detection have been published in the past [169, 170, 171]. As to date, most studies in the PCD field focus on improving technical parameter analysis using state of the art algorithms. The challenges posed in this field by the always-growing complexity of WDS calls for original, novel thinking and methodologies. As opposed to the PCD scheme described above, an Active Contamination Detection (ACD) scheme considers deliberate reconfiguration of system components and hydraulics to achieve better detectability. ACD is based on a general concept in the fault-diagnosis literature termed *active fault detection*, which states that an auxiliary input — a set of known, predetermined input procedures in known times — can be applied to a system, to improve the detection ability of a given sensor set [112]. The auxiliary input is designed in a way such that specific faults trigger different responses from the system. Using a suitable fault-detection methodology, the different responses are identified and linked to a fault, thus improving fault detection time and fault isolation. Applying an *active fault detection* scheme to different types of systems is not straightforward and poses a challenging task. It has been successfully applied in the field of electrical engineering where it has been mainly studied in the past two decades [110, 111].

The field of ACD in WDS lacks previous major researches, however related work that can be considered a natural precursor of this work has been done in redesigning WDS for containment of possible contaminations [172]. In this work we focus on the motivation for generating ACD methodologies and the benefits when applied. The general problem of actively detecting a suspected contamination, while minimizing the impact of that contamination and maintaining hydraulic requirements of the network, is mathematically formulated. Then, a simplified version of the ACD problem is solved using heuristic algorithms. The proposed solution is demonstrated on two benchmark networks and the results are analyzed and discussed.

9.2 Motivational Example

The concept of ACD in WDS is demonstrated using a simple 6 (six) node network (Fig. 9.1) with arrows representing the flow direction in each of the connecting pipes between nodes. Fig. 9.1(A) describes the original state of the network with no sensors or hydraulic manipulation, the flow directions are as marked by the arrows. It can be observed from the flow directions that no single sensor location can cover the entire network. For example, when a sensor (S) is located in node 5 as in Fig. 9.1(B), node 6 is unobservable. Moving the sensor (S) to node 6 as in Fig. 9.1(C), results in the loss of observability at nodes 3 and 5. The covered nodes in each scenario are marked in gray. A full system coverage, in this layout and flow regime, cannot be achieved using only 1 (one) fixed sensor location. Fig. 9.1(D), on the other hand, shows the potential of using hydraulic manipulation in the form of pipe closure. When the connecting link between nodes 2 and 4 is closed, the flows are forced to generate a continuous path passing through all the system nodes. Therefore, if one of the nodes is infected, the contamination will surely reach the sensor at some point of time.

Moreover, if the flow regimes shown in Fig. 9.1(C) and (D) are applied alternately over the system operational times, a better source identification can be achieved. If a contamination is detected during Fig. 9.1(C) operational times, it can be located at nodes 1,2,4 or 6. If a switch to Fig. 9.1(D) causes a detection in the sensor, it can also be located at nodes 3 or 5.

It may be argued that redirecting suspected contaminating water from various parts of the network, may spread the contamination to more consumers. This is the case when the sensor is located as in Fig. 9.1(C), node 3 (three) is suspect of being contaminated and the control action of Fig. 9.1(D) is applied. This scenario results in nodes 3,4,5,6 being contaminated, while originally only nodes 3,5 would be contaminated, i.e., the contamination coverage has increased. However, a better metric of contamination impact is how much contaminated water is consumed, which is also a function of the detection time. If the alternative for confirming the existence of a contaminant is to perform manual sampling at node (3), a procedure which may take up to 24 hours, then the total contaminated water consumed by nodes 3,5 may exceed the total contaminated water consumed by nodes 3,4,5,6 when the contamination is detected in only a few hours by the sensor.

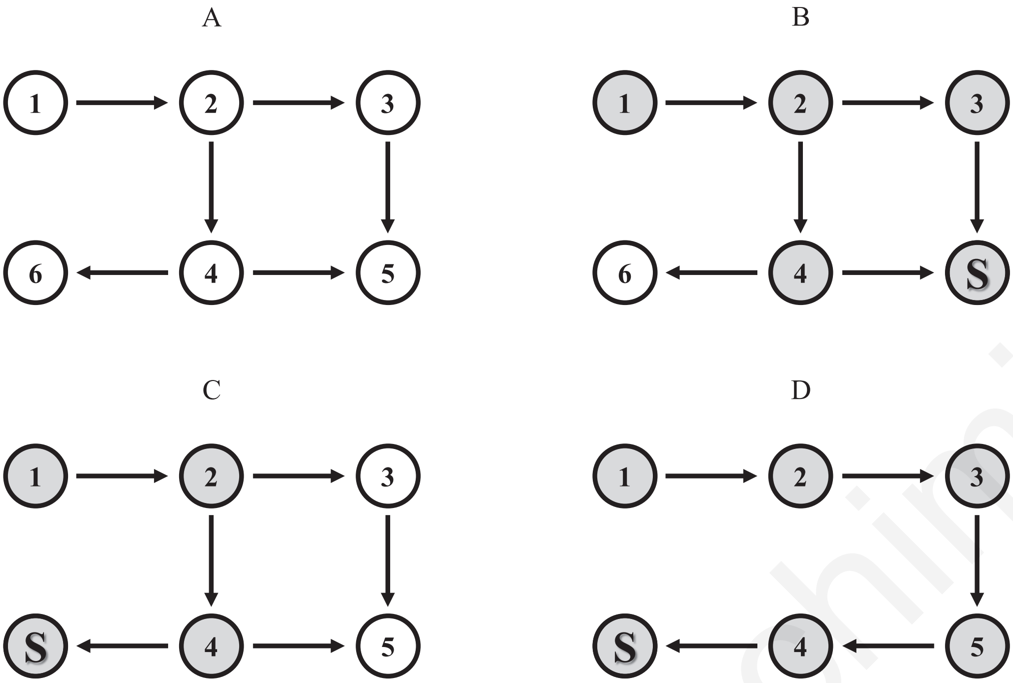


Figure 9.1. Six node system. Arrows represent flow directions, S represents fixed sensor locations and gray marks represent covered nodes.

9.3 Problem formulation

The topology of a Water Distribution Network (WDN) is described by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where the nodes $\mathcal{N}$ represent water tanks or reservoirs, junctions of pipes and water demand locations, while the links $\mathcal{L}$ represent pipes and pumps. Let the subset of nodes, indicated by $\mathcal{N}_s \subset \mathcal{N}$, represent water contamination *sensor nodes* and the subset of nodes $\mathcal{N}_c \subset \mathcal{N}$ represent *nodes suspect of contamination*, of which only one can be the source of contamination while the others represent the uncertainty of the contaminant source location.

The hydraulic state of the network is described by the flow-states $\boldsymbol{q}$, where q_j is the water flow in link $j \in \mathcal{L}$, and the head-states $\boldsymbol{h}$, where h_i is the piezometric head at node $i \in \mathcal{N}$. The piezometric head (referred to as just head from now on) consists of a component proportional to the pressure p_i at node i, and of the node elevation z_i in respect to a geodesic reference. Each node i in $\mathcal{N}$ is associated with a water demand at the node location, indicated by $d_i \in \mathbb{R}^+$.

The main hydraulic requirement in a water distribution network is that all consumer demands are satisfied, which is achieved by ensuring a defined minimum pressure $p_{\min}$ at all nodes. Additionally, a maximum pressure $p_{\max}$ should be ensured to reduce the risk of pipe failures. The main hydraulic actuators in WDN are valves which can modify flow in pipes, valves which can modify pressure at nodes and pumps which add energy in the form of pressure in the network.

Let the subset of links $\mathcal{L}_v \subset \mathcal{L}$ indicate pipes that have valves which can open or close by request. The status of pipe $j \in \mathcal{L}_v$ depends on the status of the valve on this pipe, indicated by $v_j \in \{0, 1\}$. The flow q_j in pipe j is restricted to be equal to zero when the valve is closed, i.e., when $v_j = 0$, while the flow is unrestricted when $v_j = 1$. Thus, flows in the network are dependent on the input vector $\boldsymbol{v} \in \{0, 1\}^{n_v}$, where n_v is the number of valves in the network.

Additionally, let the subset of nodes $\mathcal{N}_p \subset \mathcal{N}$ be the *pressure control nodes* in the network where pressure is regulated at a specified set-point. In this work, the head h_i of node $i \in \mathcal{N}_p$ relates to the pressure set-point indicated by u_i such that $h_i = u_i + z_i$, where z_i is the elevation of the node. These nodes represent the output of Pressure Reduction Valves (PRVs) which are usually placed at the entrances of District Metered Areas (DMAs) and their output pressure can be selected. Another way the pressure control nodes are realized, is by using pumps able of *pump head control*, i.e., the pumps are equipped with pressure sensors and are able to regulate the pressure at the pump output. The pressure set-point vector $\boldsymbol{u} \in \mathbb{R}^{n_p}$, where n_p is the number of pressure control nodes, is constrained by the physical properties of the corresponding actuating device, i.e., for PRVs the maximum pressure set-point is the PRV input pressure, and for pumps it is the pump input pressure plus the maximum pressure the pump can add. We define $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ as the lower and upper bounds respectively of the pressure set-point vector.

The hydraulic state of a WDN is calculated using the conservation of energy and mass equations [173] in discrete time, with the hydraulic step Δt corresponding to the discrete time step $k \geq 0$. The pressures and flows in the network for each time step k are calculated using the *hydraulics function* $f_h(\cdot)$ given by:

$$\begin{bmatrix} \boldsymbol{p}(k) \\ \boldsymbol{q}(k) \end{bmatrix} = f_h\left(\boldsymbol{d}(k), \boldsymbol{h}(k-1), \boldsymbol{u}(k), \boldsymbol{v}(k); \mathcal{G}\right) \tag{9.1}$$

This work aims to develop an algorithm which enables the water quality monitoring of the nodes in $\mathcal{N}_c$, using the available stationary sensors in $\mathcal{N}_s$ by manipulating the valve $\boldsymbol{v}$ and pressure $\boldsymbol{u}$ settings in the network. Monitoring a node in $\mathcal{N}_c$ is defined as the event when water originating from that node passes through any of the sensor nodes in $\mathcal{N}_s$ at a later time step. Specifically, let $f_{tr}(\cdot)$ be defined as the *contaminant trace function*, which tracks the spread of a contaminant from a node $i \in \mathcal{N}_c$ to other nodes in $\mathcal{N}$. The set of contaminated nodes due to node i at time step k is denoted by $\mathcal{N}_c^i(k)$ and given by:

$$\mathcal{N}_c^i(k) = f_{tr}\left(\boldsymbol{Q}(k), \boldsymbol{V}(k), i \in \mathcal{N}_c\,; \mathcal{G}\right), \tag{9.2}$$

where $\boldsymbol{Q}(k)$ is the sequence of flow vectors such that $\boldsymbol{Q}(k) = [\boldsymbol{q}(1), \boldsymbol{q}(2), ..., \boldsymbol{q}(k)]$ and $\boldsymbol{V}(k)$ is the sequence of valve control vectors such that $\boldsymbol{V}(k) = [\boldsymbol{v}(1), \boldsymbol{v}(2), ..., \boldsymbol{v}(k)]$. Finally, monitoring of a node $i \in \mathcal{N}_c$ is achieved when $\mathcal{N}_c^i(k) \cap \mathcal{N}_s \neq \emptyset$. The earliest time step at which a node $i \in \mathcal{N}_c$ has been monitored is defined as the *detection time step* and is given by $k_d^i = \min\left\{k : \mathcal{N}_c^i(k) \cap \mathcal{N}_s \neq \emptyset\right\}, \forall i \in \mathcal{N}_c$.

It is necessary to define a maximum allowable detection time step, denoted by $\bar{k}_d$. In addition, the set $\mathcal{K} = \{1, \cdots, \bar{k}_d\}$ is defined as the set of allowable time steps such that only when $k_d \in \mathcal{K}$ is a detection considered successful. The reason for defining $\bar{k}_d$ may be to ensure that a contaminant will reach a sensor location in a detectable concentration even if it decays in the water. However, it can also be determined using other time constraints defined by the utility operator. In some cases, depending on the available sensors or network topology, it may be preferable to perform manual sampling instead of redirecting the contaminant to a sensor. The decision of whether to perform manual sampling depends on the estimated time it would take to perform this procedure. This information is incorporated into the maximum detection time step $\bar{k}_d$ of the proposed ACD methodology.

The primary objective of the proposed methodology is to minimize the impact of any suspected contamination. In this work, for simplicity, the *contamination impact* refers to the volume of contaminated water consumed and is calculated using the impact formula $f_{imp}(\cdot)$. The formula can be adapted to include different aspects of the "damage" caused by contaminant consumption [174]. It is assumed that when detection is confirmed, preventative actions immediately take place that prevent further contamination. The contamination impact at time step k of a contaminant originating from node $i \in \mathcal{N}_c$, is denoted by $I^i(k)$ and calculated as follows:

$$I^i(k) = f_{imp}\left(k_d^i, \boldsymbol{D}(k); \mathcal{N}_c^i(k)\right) = \begin{cases} \Delta t \sum\limits_{\tau=1}^{k_d^i} \sum\limits_{j \in \mathcal{N}_c^i(\tau)} d_j(\tau),\; k_d^i \leqslant k \\ \Delta t \sum\limits_{\tau=1}^{k} \sum\limits_{j \in \mathcal{N}_c^i(\tau)} d_j(\tau),\; k_d^i > k \end{cases} \tag{9.3}$$

where $\boldsymbol{D}(k)$ is the sequence of demand vectors such that $\boldsymbol{D}(k) = [\boldsymbol{d}(1), \boldsymbol{d}(2), ..., \boldsymbol{d}(k)]$ and $\mathcal{N}_c^i(\tau)$ is the set of contaminated nodes in the network at time step τ due to contaminants originating from node $i \in \mathcal{N}_c$. Due to unknown source of contamination, it is necessary to calculate the impact from a contaminant originating from any of the nodes in $\mathcal{N}_c$. The final impact is the maximum of the possible impacts, calculated as follows:

$$I(k) = \max_i \left\{ f_{imp} \left(k_d^i, \boldsymbol{D}(k), \mathcal{N}_c^i(k) \right) \right\}, \forall i \in \mathcal{N}_c \tag{9.4}$$

As a secondary objective, the proposed methodology minimizes the valve control actions, as it may be infeasible to close/open a large number of valves, especially when these are not remotely controlled. The number of valve control actions taken at time step k can be calculated by taking the L^1-norm of the difference $\left| \boldsymbol{v}(k) - \boldsymbol{v}(k-1) \right|_1$. Let $\Delta V(k)$ indicate the total number of valve control actions taken until time step k, and calculated as follows:

$$\Delta V(k) = \sum_{\tau=1}^{k} \left| \boldsymbol{v}(\tau) - \boldsymbol{v}(\tau - 1) \right|_1. \tag{9.5}$$

Closing valves can significantly affect the pressures in the network. The methodology should ensure that a hydraulic solution within the pre-defined pressure requirements is feasible. This can be achieved by letting the algorithm choose the set-points $\boldsymbol{u}(k)$ for pressure controlled nodes, for which the corresponding pressure constraints should also apply.

We then define the following multi-objective optimization problem:

Problem 1. *Given a water network with a set of sensor nodes $\mathcal{N}_s$ and a set of nodes suspect of contamination $\mathcal{N}_c$ at time $k = 0$, find the valve control vector sequence $\boldsymbol{V}(\bar{k}_d) = [\boldsymbol{v}(1), \boldsymbol{v}(2), ..., \boldsymbol{v}(\bar{k}_d)]$ and the pressure set-point sequence $\boldsymbol{U}(\bar{k}_d) = [\boldsymbol{u}(1), \boldsymbol{u}(2), ..., \boldsymbol{u}(\bar{k}_d)]$ such that:*

$$\begin{aligned}
\underset{\boldsymbol{V}(\bar{k}_d), \boldsymbol{U}(\bar{k}_d)}{\operatorname{argmin}} \quad & \left\{ I(\bar{k}_d), \Delta V(\bar{k}_d) \right\} \\
& I(\bar{k}_d) = \max_i \left\{ f_{imp} \left(k_d, \boldsymbol{D}(\bar{k}_d), \mathcal{N}_c^i(k) \right) \right\}, \forall i \in \mathcal{N}_c \\
& k_d^i = \min \left\{ k \in \mathcal{K} : \mathcal{N}_s \cap \mathcal{N}_c^i(k) \neq \emptyset \right\}, \forall i \in \mathcal{N}_c \\
s.t. \quad & \Delta V(\bar{k}_d) = \sum_{\tau=1}^{\bar{k}_d} \left| \boldsymbol{v}(\tau) - \boldsymbol{v}(\tau - 1) \right|_1 \\
& [\boldsymbol{p}(k)^\top \, \boldsymbol{q}(k)^\top]^\top = f_h \left(\boldsymbol{d}(k), \boldsymbol{h}(k-1), \boldsymbol{u}(k), \boldsymbol{v}(k); \mathcal{G} \right), \forall k \in \mathcal{K} \\
& p_{\min} \leq \boldsymbol{p}(k) \leq p_{\max}, \ \forall k \in \mathcal{K} \\
& \boldsymbol{u}_{\min} \leq \boldsymbol{u}(k) \leq \boldsymbol{u}_{\max}, \ \forall k \in \mathcal{K},
\end{aligned}$$

Problem 1 is a multi-objective optimization problem and due to its high complexity, some simplifications need to be made to solve it. In the following sections a simplified version of Problem 1 is defined and then a solution methodology using heuristic optimization algorithms is given.

9.4 Problem simplification

The nonlinear functions in Problem 1 require the knowledge of water demands at each time step and the initial conditions in the network in order to be calculated. The following assumption is therefore imposed:

Assumption 9.4.1. *Water demands* $\boldsymbol{d}(k)$ *in the network are known for* $k \in \mathcal{K}$ *and change only at the defined discrete time steps. The initial head-state conditions of the network* $\boldsymbol{h}(0)$ *are known.*

The valve settings $\boldsymbol{v}(k)$ are considered as binary variables. Depending on the number of valves $n_v = |\mathcal{L}_v|$, the number of combinations of this input for each time step is 2^{n_v}. Considering that valve settings can change at every time step, then for the valve control sequence $\boldsymbol{V}(\bar{k}_d)$ there are $2^{n_v \times \bar{k}_d}$ possible combinations. This search space is large even for heuristic algorithms to handle and in the case of large networks a solution may never be found. The problem can be simplified by reducing the decision variables search space using the following assumption:

Assumption 9.4.2. *Only one control input is applied to the system, such that the valve control vector* $\boldsymbol{v} = \boldsymbol{v}(1) = \boldsymbol{v}(2) = \cdots = \boldsymbol{v}(\bar{k}_d)$ *and the pump set-point vector* $\boldsymbol{u} = \boldsymbol{u}(1) = \boldsymbol{u}(2) = \cdots = \boldsymbol{u}(\bar{k}_d)$.

The pressure set-points $\boldsymbol{u}$ are continuous variables, which are bounded due to physical and actuator limitations. Problem 1 can be further simplified by discretization of the pressure set-points using the following assumption:

Assumption 9.4.3. *Pressure set-points are discrete and predefined for each node* $i \in \mathcal{N}_p$, *such that* $u_i \in \mathcal{U}$, *where the set of pressure settings* $\mathcal{U} = \{u_{i,\min}, \cdots, u_{i,\max}\}$ *has finite elements.*

The objective function of Problem 1 can be simplified by defining a single objective which is the linear combination of impact $I(k)$ and number of valve actions $\Delta V(k)$. In order to perform this simplification, the two quantities must first be normalized and then given appropriate weights. Let $I_{max}(k)$ denote the impact upper bound at time step k which is equal to the total water consumed in the network until time k, the number of valves n_v be the maximum number of valve actions when Assumption 9.4.2 holds and $\beta \in [0, 1]$ be an appropriately chosen weight factor. The single-objective function cost will then be given by:

$$J(k) = (\beta / I_{max}(k))\, I(k) + ((1-\beta)/n_v)\, \Delta V(k). \tag{9.6}$$

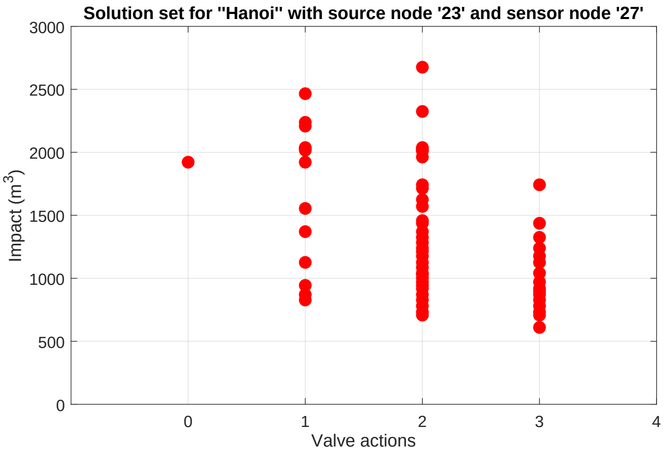


Figure 9.2. Feasible solutions illustrating the Pareto frontier for Hanoi network with one sensor

The conversion of this multi-objective problem into a single-objective problem using the weighted sum of two criteria, has the advantage of reduced computational complexity and the disadvantage of losing some solutions which may be optimal or near-optimal, and would appear on the 2D Pareto frontier. The selection of parameter β is important, in order to extract the best solution considering the defined optimality criteria. In this work β was selected by the authors so that the impact objective receives a higher priority. The shape of the Pareto frontier, depends on the network topology and the feasibility of hydraulic solutions when the algorithm tries to construct a path from the contamination source to the sensor location. Networks with looped topology may allow a larger number of valves to be closed, thus providing more freedom in constructing paths which will result in smaller detection times and less impact. An example of the Pareto frontier created for the specific case of the Hanoi network when the contaminant source is at node ‘23’ and the sensor is at node ‘27’, creates a convex Pareto Front, as depicted in Fig 9.2. Note that feasible hydraulic solutions for this case exist only for 0-3 valve actions.

We then define a simplified version of Problem 1 as follows:

Problem 2. *Given a water network with a set of sensor nodes $\mathcal{N}_s$ and a set of nodes suspect of contamination $\mathcal{N}_c$ at time $k = 0$, find the valve control vector $\boldsymbol{v}$ and the pressure set-point* *vector $\boldsymbol{u}$ such that:*

$$\begin{aligned}
\underset{\boldsymbol{v},\boldsymbol{u}}{\text{argmin}} \quad & J(\bar{k_d}) = \Big(\beta / I_{max}(\bar{k_d})\Big) I(\bar{k_d}) + ((1-\beta)/n_v)\, \Delta V(\bar{k_d}). \\
& I(\bar{k_d}) = \max_i \left\{ f_{imp}\left(k_d^i, \boldsymbol{D}(\bar{k_d}); \mathcal{N}_c^i(k)\right)\right\}, \forall i \in \mathcal{N}_c \\
& k_d^i = \min\left\{k \in \mathcal{K} : \mathcal{N}_s \cap \mathcal{N}_c^i(k) \neq \emptyset\right\}, \forall i \in \mathcal{N}_c \\
s.t. \quad & \Delta V(\bar{k_d}) = \left|\boldsymbol{v}(1) - \boldsymbol{v}(0)\right|_1, \\
& [\boldsymbol{p}(k)^\top \ \boldsymbol{q}(k)^\top]^\top = f_h\left(\boldsymbol{d}(k), \boldsymbol{h}(k-1), \boldsymbol{u}, \boldsymbol{v}; \mathcal{G}\right), \forall k \in \mathcal{K} \\
& p_{\min} \leq \boldsymbol{p}(k) \leq p_{\max}, \ \forall k \in \mathcal{K} \\
& \boldsymbol{u} \in \mathcal{U}
\end{aligned}$$

9.5 Solution using Evolutionary Algorithms

In this work we propose a solution methodology which uses a Genetic Algorithm (GA) implemented by MATLAB® to select the appropriate inputs $\boldsymbol{v}$ and $\boldsymbol{u}$ which minimize the objective function of Problem 2. Additionally, it uses a hydraulic simulator to implement the hydraulic solver of (9.1) and contaminant trace function of (9.2) for a given network. The solution methodology is described below.

9.5.1 Implementation of nonlinear functions

The EPANET water distribution network simulator libraries [74] are used with the EPANET-MATLAB Toolkit as described in Appendix A′ to implement the hydraulic solver of (9.1) and contaminant trace function of (9.2) for a given network. The contaminant trace function $f_{tr}(\cdot)$ provided by EPANET, returns the set of nodes that are affected by the contaminant, as well as the percentage of contaminated water arriving at each node. A threshold c_{thr} is defined to indicate the minimum percentage of contaminant concentration that sensors can detect.

9.5.2 Decision variables

The GA decision variables are the vector of valve input $\boldsymbol{v}$ and the vector of pressure set-points $\boldsymbol{u}$. These are given to the GA as a single vector $[\boldsymbol{v}^\top \boldsymbol{u}^\top]^\top$ and later used appropriately in the hydraulic simulation.

9.5.3 Fitness function

The GA evaluates each input using an appropriately selected fitness function. In this work, the fitness function represents the objective function of Problem 2. In addition, it includes terms that penalize solutions that do not lie in the search space defined by the problem constraints. This is a way to impose these constraints which also assists the GA to find a feasible solution.

The first term of the fitness function is the pressure constraint penalty which penalizes pressure deviations outside the pressure constraints. Given the inputs $\boldsymbol{v}$ and $\boldsymbol{u}$, a hydraulic simulation in EPANET calculates the pressures $\boldsymbol{h}(k), k \in \mathcal{K}$ in the network as in (9.1). The pressure penalty is then defined as:

$$P = \max_{k\in\mathcal{K}}\left(\max_{i\in\mathcal{N}}\left(p_{min} - p_i(k), 0\right)\right) + \max_{k\in\mathcal{K}}\left(\max_{i\in\mathcal{N}}\left(p_i(k) - p_{\max}, 0\right)\right) \tag{9.7}$$

After calculating the hydraulics of the network, the EPANET contaminant trace function calculates the set of contaminated nodes in the network $\mathcal{N}_c(k), \forall k \in \mathcal{K}$ as in (9.2). The contaminant detection time k_d^i for each node $i \in \mathcal{N}_c$ is then calculated as follows:

$$k_d^i = \begin{cases} \min\left\{k \in \mathcal{K} : \mathcal{N}_c^i(k) \cap \mathcal{N}_s \neq \emptyset\right\}, & \mathcal{N}_c^i(k) \cap \mathcal{N}_s \neq \emptyset \\ \bar{k}_d + 1\,, & \mathcal{N}_c^i(k) \cap \mathcal{N}_s = \emptyset \end{cases} \tag{9.8}$$

Using the previous results, the contamination impact $I(\bar{k}_d)$ is calculated as follows:

$$I(\bar{k}_d) = \max_i\left\{f_{imp}\left(k_d^i, \boldsymbol{D}(\bar{k}_d), \mathcal{N}_c^i(\bar{k}_d)\right)\right\}, \forall i \in \mathcal{N}_c \tag{9.9}$$

The cost $J_{GA}(\bar{k}_d)$ of the GA fitness function is then given by:

$$J_{GA}(\bar{k}_d) = \alpha P + \left(\beta / I_{max}(\bar{k}_d)\right) I(\bar{k}_d) + \left((1-\beta)/n_v\right) \Delta V(\bar{k}_d). \tag{9.10}$$

where $\alpha >> \beta$ is the weight factor for the pressure constraint term.

The 'pressure constraint penalty' can be incorporated into a custom GA selection function, thus removing it from the cost function $J_{GA}(\bar{k}_d)$. This has the added benefit of having one less weight factor to tune (α). However, the selection function should be constructed in such way that maintains the property of pointing the search direction towards solutions that do not violate pressure constraints by quantifying how "far" the current solution is from a feasible solution. This is why the selection function should have as input, not only the current solution cost, but also the "pressure constraint penalty" as defined in (9.7).

9.6 Example applications

9.6.1 Improving contamination monitoring example

The benchmark network "Hanoi" is used to demonstrate a solution to Problem 2 using the proposed methodology described in the previous section. The network is simulated for 24

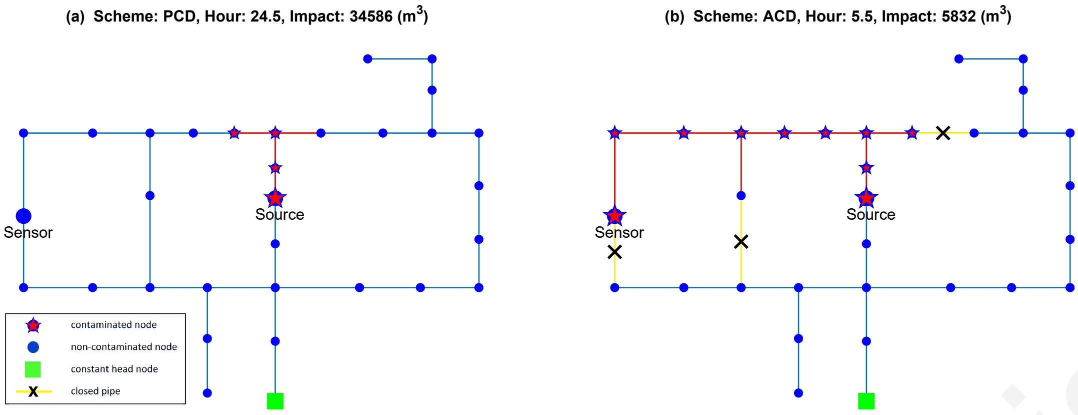

(a) Scheme: PCD, Hour: 24.5, Impact: 34586 (m^3)
(b) Scheme: ACD, Hour: 5.5, Impact: 5832 (m^3)
Sensor
Source
Sensor
Source
contaminated node
non-contaminated node
constant head node
closed pipe

Figure 9.3. Example of monitoring a previously unmonitored node and reducing the contamination impact on the Hanoi network: (a) Contaminant never reaches the sensor in PCD scheme, (b) Contaminant reaches the sensor when applying the proposed ACD methodology.

hours, with a hydraulic time step of $\Delta t = 30$ minutes, using realistic water demand patterns. The maximum detection time step is set to $\bar{k}_d = 48$ time steps, which is equivalent to 24 hours. The network model used is available in Supplemental Data.

In the example of Fig. 9.3, we demonstrate a case in which a previously unmonitored node is able to be monitored using the proposed scheme. The sensor is placed at node $\mathcal{N}_s = \{30\}$ and the node suspect of contamination is $\mathcal{N}_c = \{18\}$, illustrated with blue and red colors respectively. The contamination spread is indicated with red color. Note that in the figure only an instance of the simulation is shown, as the hydraulic state of the network changes at each time step. Fig. 9.3(a) shows the maximum contaminant spread in the default PCD scheme where all the pipes are open. In this scenario, the contaminant never reaches the sensor. The application of the proposed ACD methodology on this scenario, yields a valve control vector $\boldsymbol{v}$ which closes the three pipes illustrated in Fig. 9.3(b). By closing these pipes, a flow path between the contamination source and sensor is created and sustained long enough for the contaminant to reach the sensor. The pressure control input $\boldsymbol{u}$ changes the head of the pressure controlled node from 100 (m) to 190 (m) in order to avoid negative pressures.

An example on the same network is given in Fig. 9.4, which illustrates a case where the proposed algorithm reduces the monitoring time of a contaminated node. The sensor node is selected to be $\mathcal{N}_s = \{30\}$ and the node suspect of contamination is $\mathcal{N}_c = \{1\}$. Fig. 9.4(a) shows that in the PCD scheme, where all pipes are open, the contaminant reaches the sensor at $k_d = 20$ time steps, or 10 hours. The application of the proposed methodology on this scenario, yields a control vector $\boldsymbol{v}$ which indicates that pipes “28” and “30” highlighted in

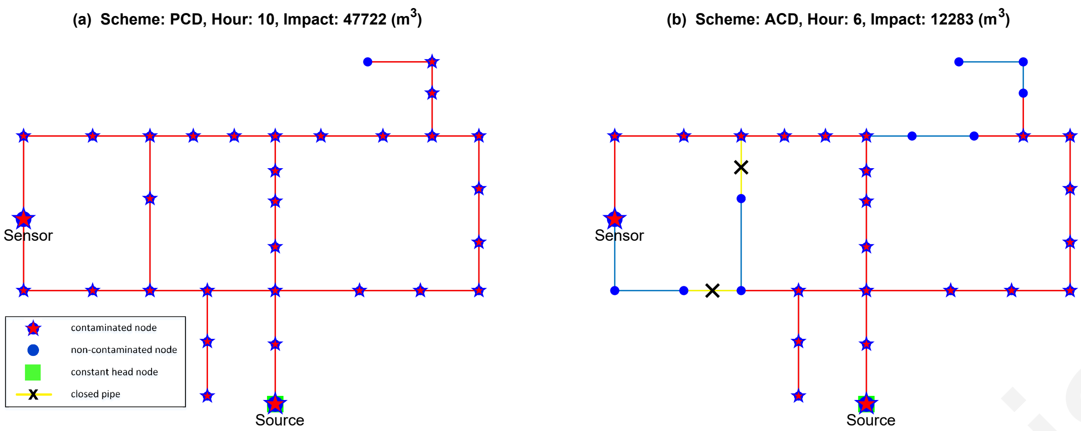

(a) Scheme: PCD, Hour: 10, Impact: 47722 (m^3)
Sensor
Source
contaminated node
non-contaminated node
constant head node
closed pipe
(b) Scheme: ACD, Hour: 6, Impact: 12283 (m^3)
Sensor
Source

Figure 9.4. Example of monitoring time and contamination impact reduction on the Hanoi network: (a) Contaminant detection time in PCD scheme, (b) Contaminant detection time when applying the proposed ACD methodology.

Fig. 9.4(b) should be closed. Using this configuration the detection time is reduced to $k_d = 18$ (time steps), which is equivalent to 9 hours. The ACD scheme thus managed to reduce the contaminant detection time by one hour by closing two valves.

9.6.2 Multiple suspected nodes example

The ability of the algorithm to handle multiple suspected contamination nodes is demonstrated in this section with two examples on the "Hanoi" network, as described in the previous section. In the example of Fig. 9.5, we demonstrate the same two instances of the Hanoi network as in Fig. 9.3 and Fig. 9.4, with the difference that in these example a group of nodes is suspect of contamination instead of only one node. In the first instance (Fig. 9.5(a)), the sensor is placed at node $\mathcal{N}_s = \{30\}$, the nodes suspect of contamination is the set $\mathcal{N}_c = \{15, 16, 17, 18, 27\}$ (orange color) and the actual contamination occurs at node $\{18\}$ (black circle). In the second instance (Fig. 9.5(b)), the sensor is placed at node $\mathcal{N}_s = \{30\}$, the nodes suspect of contamination is the set $\mathcal{N}_c = \{1, 2, 3, 20\}$ (orange color) and the actual contamination occurs at node $\{1\}$ (black circle). The contamination spread is indicated with red color. As illustrated in Fig. 9.5, the algorithm calculates a flow path to the sensor which includes all suspected nodes, thus monitoring them all. Notice that the solutions differ from the examples in Fig. 9.3 and 9.4, as the algorithm tries to monitor all the suspected nodes. The contamination impact calculated is the maximum possible impact that would have occurred if any one of the suspected nodes was the source of contamination.

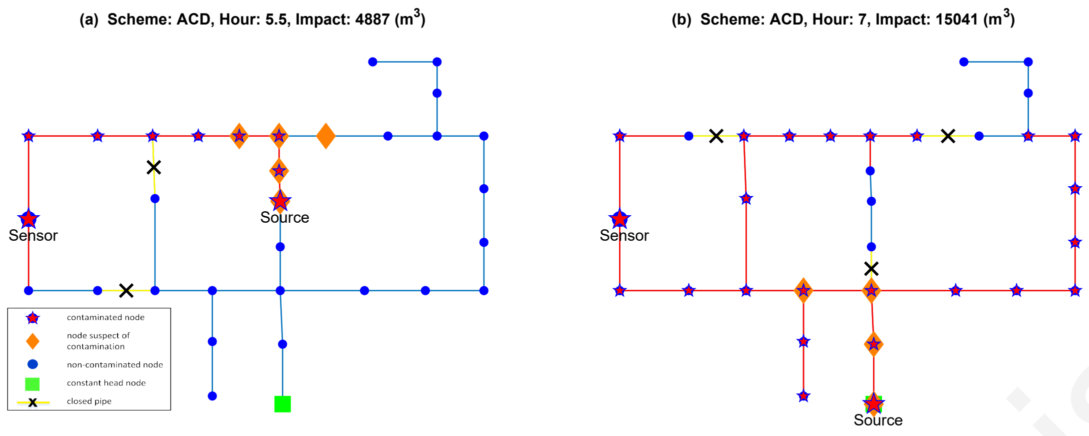

(a) Scheme: ACD, Hour: 5.5, Impact: 4887 (m3)
Sensor
Source
contaminated node
node suspect of contamination
non-contaminated node
constant head node
closed pipe
(b) Scheme: ACD, Hour: 7, Impact: 15041 (m3)
Sensor
Source

Figure 9.5. Example of multiple nodes being suspect of contamination: (a) The ACD solution to the problem of Fig. 2 with 5 suspected nodes, (b) The ACD solution to the problem of Fig. 3 with 4 suspected nodes.

9.6.3 Transport Network example

In this case study we examine the benefits of using the ACD scheme in a network with optimally placed sensors. We compare the contamination impact and detection times in the default PCD case and when using the ACD scheme, for all possible contamination scenarios in the network, i.e., for every possible contamination source (nodes). The "Hanoi" network is used, which is a representative of a transport network with large pipes and demands.

Contaminant sensors were optimally placed using the *S-Place* toolkit [31]. The *S-Place* toolkit places the sensors by minimizing the impact of all possible contaminations. Impact in *S-Place* is defined as the volume of contaminated water consumed in m^3 until the contamination is detected, similar to the definition in this work. The sets of 1,2 and 3 sensors are placed using exhaustive search, for a simulation period lasting 24 hours, deliberately chosen to match the maximum detection time $\bar{k}_d$. The toolkit also has the option to consider demand and parameter uncertainty during the sensor placement process, however it was not used because we assume known model and demands in this work.

Exhaustive search simulations were performed, where the following parameters were varied:

1. The number of sensors. Three placement scenarios were considered: 1, 2 and 3 optimally placed sensors, with sensors placed at nodes $\mathcal{N}_s = \{27\}$, $\mathcal{N}_s = \{11, 27\}$ and $\mathcal{N}_s = \{11, 21, 27\}$ respectively.
2. Contamination source. Contamination originating from all the nodes in the network (only one contamination node at each simulation).

3. Contamination detection scheme. Passive (PCD) or Active (ACD) contamination detection scheme is applied at each simulation.

The proposed ACD scheme manipulates valves and the reservoir head in order to drive the contamination to a sensor, while satisfying hydraulic constraints. It assumes that the contamination node is known *a priori*. The objective is to minimize the contamination impact and valve control actions. The constraints when the ACD scheme is applied are defined as follows:

1. Pressure constraints for all nodes are set to $p_{\min} = 20\,(m)$ and $p_{\max} = 150\,(m)$.
2. Concentration at sensors must be greater than $c_{thr} = 7\%$ for detection.

For the selection of minimum pressure, $p_{\min} = 20\,(m)$ was used because it is a common pressure specification of water utilities during fire flow [175]. For the maximum pressure, it is considered that the Hanoi network is a transport network with large pipes which can withstand pressures up to $150\,(m)$, as the demands are DMA entrances.

The threshold c_{thr} is the minimum percentage of contaminant concentration that a sensor can detect. The maximum percentage is 100% and corresponds to the maximum concentration that same sensor can detect. The “contaminant” can be any water quality parameter of interest. In general, the choice of the threshold does not affect the operation of the algorithm and is problem-specific. To demonstrate how the threshold c_{thr} can relate to real water quality parameters, the authors consider the so-called indicator parameter of *color*, which is measured in *Hazen* units. The motivation behind this choice is the fact that pipe discoloration is a common problem in the drinking water distribution systems of countries with water shortages, due to changes in water chemistry resulting from the increasing use of desalinated water. The selection of a threshold takes into account the measurement range $(0 - 200)$ Hazens of an industrial sensor, e.g., the “i::scan NTU/FTU+Color” multi-parameter spectrophotometer probe. The maximum allowable color in drinking water ranges from 15-20 Hazens (e.g., see European Water Quality Standards). These values correspond approximately to the selected 7% threshold $(15/200)$ considering the sensor measurement range.

As a technical note, the procedure of confirming the existence of a contaminant does not need additional information when specialized sensors which detect specific types of contaminants are used. However, indicator-parameter sensors (chlorine concentration, conductivity, Oxidation Reduction Potential (ORP), etc.) are commonly used in WDS and will only indicate a variation in a specific water quality parameter. An appropriate threshold is then needed, which will be calculated by an event detection algorithm in order to confirm the presence of

	Cov (%)		$\tilde{k_d}$ (hours)		$\tilde{I}$ (m^3)		$\tilde{VC}$	$\tilde{H}$ (m)
	PCD	ACD	PCD	ACD	PCD	ACD	ACD	ACD
Hanoi with 2 sensors								
$c_{thr} = 4\%$	65.6	78.1	5.5	2.5	4962	1199	2	105
$c_{thr} = 7\%$	65.6	71.9	5.5	2.5	4962	1203	2	100
$c_{thr} = 10\%$	65.6	71.9	5.5	2.5	5104	1334	2	105
CY-DMA with 2 sensors								
$c_{thr} = 4\%$	64.8	80.2	4	1	21	3	6	40
$c_{thr} = 7\%$	61.5	79.1	4	1	24	3	6	40
$c_{thr} = 10\%$	58.3	80.2	4	1	30	3	6	40

c_{thr}	Minimum detectable contaminant concentration in (%)
Cov:	Percentage of nodes monitored by sensors.
$\tilde{k_d}$:	Median contamination detection time in hours.
$\tilde{I}$:	Median contamination impact in m^3.
$\tilde{VC}$:	Median number of valves closed by the ACD scheme.
$\tilde{H}$:	Median head at pressure control nodes in meters.

Table 9.1. Result metrics from simulation scenarios using Passive (PCD) and Active (ACD) contamination detection schemes on Hanoi and CY-DMA networks with 2 sensors and varying detection thresholds 4%, 7% and 10%.

a contaminant. When using the ACD methodology, an event detection algorithm should be able to anticipate the change in water quality due to the alteration of water flows. This is why a model-based event detection algorithm is recommended to be used with the ACD methodology which is able to incorporate into the detection logic the hydraulic changes made by the ACD methodology.

Simulation results include the Figures 9.6 to 9.8, that compare the default PCD scheme (a) and the ACD scheme (b) for each sensor placement case. The contamination source nodes that have been monitored are highlighted in yellow and the detection time and contamination impact are shown above and below each node respectively. The results of this case study are discussed in the section “Discussion of Results”.

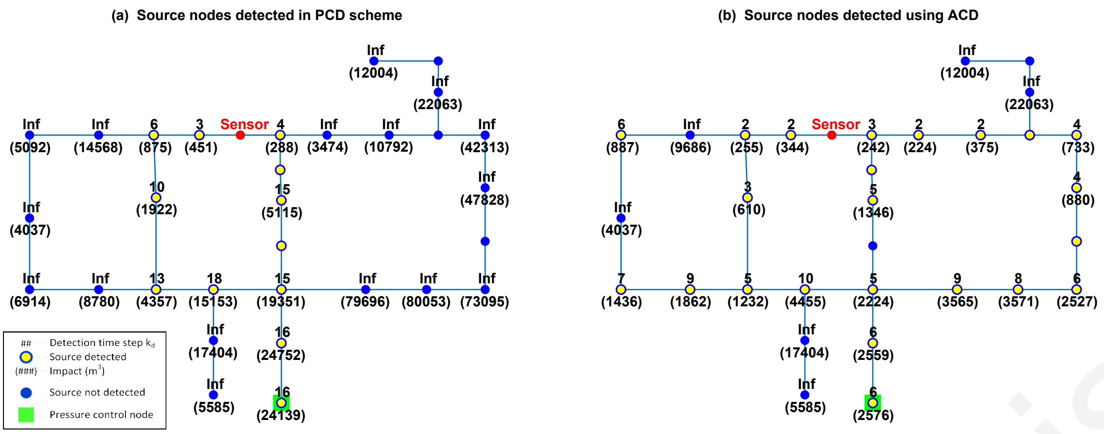

(a) Source nodes detected in PCD scheme
Inf (12004)
Inf (22063)
Inf (5092) Inf (14568) 6 (875) 3 (451) Sensor 4 (288) Inf (3474) Inf (10792) Inf (42313)
10 (1922) 15 (5115) Inf (47828)
Inf (4037)
Inf (6914) Inf (8780) 13 (4357) 18 (15153) 15 (19351) Inf (79696) Inf (80053) Inf (73095)
Inf (17404) 16 (24752)
Inf (5585) 16 (24139)
## Detection time step k_d
Source detected
(###) Impact (m³)
Source not detected
Pressure control node
(b) Source nodes detected using ACD
Inf (12004)
Inf (22063)
6 (887) Inf (9686) 2 (255) 2 (344) Sensor 3 (242) 2 (224) 2 (375) 4 (733)
3 (610) 5 (1346) 4 (880)
Inf (4037)
7 (1436) 9 (1862) 5 (1232) 10 (4455) 5 (2224) 9 (3565) 8 (3571) 6 (2527)
Inf (17404) 6 (2559)
Inf (5585) 6 (2576)

Figure 9.6. Hanoi network with 1 optimally placed sensor: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

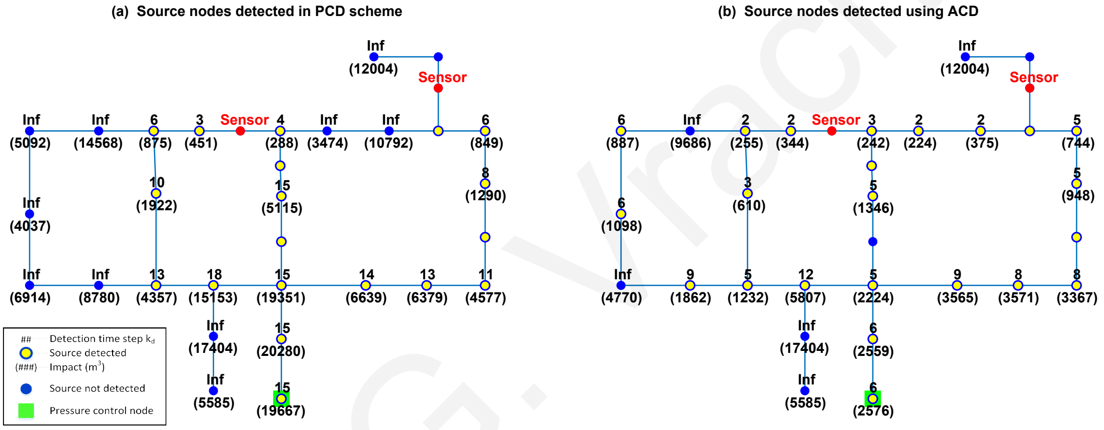

(a) Source nodes detected in PCD scheme
Inf (12004)
Sensor
Inf (5092) Inf (14568) 6 (875) 3 (451) Sensor 4 (288) Inf (3474) Inf (10792) 6 (849)
10 (1922) 15 (5115) 8 (1290)
Inf (4037)
Inf (6914) Inf (8780) 13 (4357) 18 (15153) 15 (19351) 14 (6639) 13 (6379) 11 (4577)
Inf (17404) 15 (20280)
Inf (5585) 15 (19667)
## Detection time step k_d
Source detected
(###) Impact (m³)
Source not detected
Pressure control node
(b) Source nodes detected using ACD
Inf (12004)
Sensor
6 (887) Inf (9686) 2 (255) 2 (344) Sensor 3 (242) 2 (224) 2 (375) 5 (744)
3 (610) 5 (1346) 5 (948)
6 (1098)
Inf (4770) 9 (1862) 5 (1232) 12 (5807) 5 (2224) 9 (3565) 8 (3571) 8 (3367)
Inf (17404) 6 (2559)
Inf (5585) 6 (2576)

Figure 9.7. Hanoi network with 2 optimally placed sensors: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

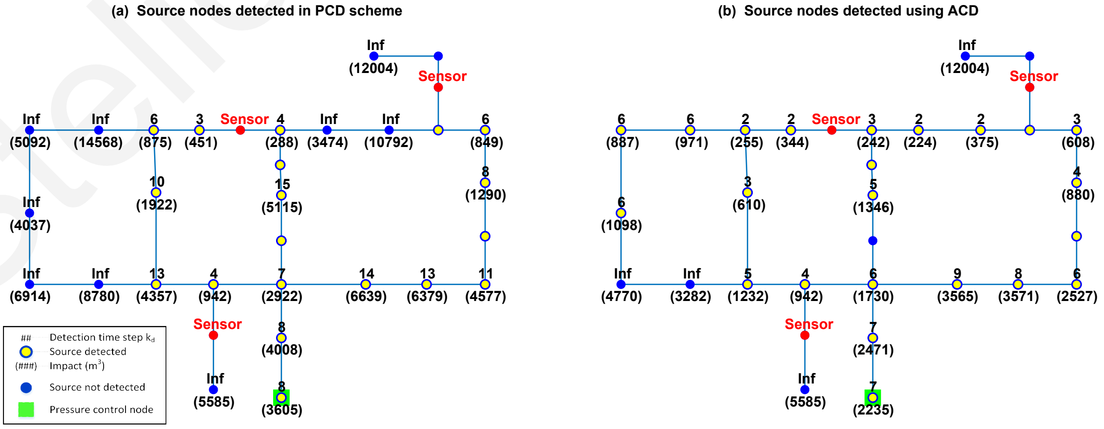

(a) Source nodes detected in PCD scheme
Inf (12004)
Sensor
Inf (5092) Inf (14568) 6 (875) 3 (451) Sensor 4 (288) Inf (3474) Inf (10792) 6 (849)
10 (1922) 15 (5115) 8 (1290)
Inf (4037)
Inf (6914) Inf (8780) 13 (4357) 4 (942) 7 (2922) 14 (6639) 13 (6379) 11 (4577)
Sensor 8 (4008)
Inf (5585) 8 (3605)
## Detection time step k_d
Source detected
(###) Impact (m³)
Source not detected
Pressure control node
(b) Source nodes detected using ACD
Inf (12004)
Sensor
6 (887) 6 (971) 2 (255) 2 (344) Sensor 3 (242) 2 (224) 2 (375) 3 (608)
3 (610) 5 (1346) 4 (880)
6 (1098)
Inf (4770) Inf (3282) 5 (1232) 4 (942) 6 (1730) 9 (3565) 8 (3571) 6 (2527)
Sensor 7 (2471)
Inf (5585) 7 (2235)

Figure 9.8. Hanoi network with 3 optimally placed sensors: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

9.6.4 District Metered Area example

In this case study, a realistic network from a large water utility in Cyprus is used. The "CY-DMA" network represents a District Metered Area of which, as opposed to the "Hanoi" network, the pipes are of smaller diameter and demands are residential consumers. Additionally it has a more complex structure, thus demonstrating the ability of the proposed methodology to perform in more realistic complex situations.

The scenario variations and constraints are the same as described in section "Transport Network example", with the exception of the sensor locations (selected using the *S-Place* toolkit) and the pressure constraints. Three placement scenarios are considered: 1, 2 and 3 optimally placed sensors, with sensors placed at nodes $\mathcal{N}_s = \{26\}$, $\mathcal{N}_s = \{9, 28\}$ and $\mathcal{N}_s = \{9, 28, 36\}$ respectively. Due to this being a network with residential consumers, in order to protect household piping from high pressures the maximum pressure constraint was reduced to the recommended value of $p_{\max} = 80\ (m)$ [175].

The results include the Figures 9.12 to 9.14, that compare the PCD scheme (a) and the ACD scheme (b) for each sensor placement case. The results of this case study are discussed in the section "Discussion of Results".

9.7 Additional network example: M1

In this case study, the "M1" network is used. The scenario variations and constraints are the same as Example 1: Hanoi. Three placement scenarios are considered: 1, 2 and 3 optimally placed sensors, with sensors placed at nodes $\mathcal{N}_s = \{26\}$, $\mathcal{N}_s = \{9, 28\}$ and $\mathcal{N}_s = \{9, 28, 36\}$ respectively. The results on "Example 2: M1", do not indicate an increase in coverage. This is due to the looped topology of this network. However, contamination impact is again reduced significantly, seeing a reduction in median contamination impact by five times in the one sensor case. Simulation times are increased compared to the Hanoi case and range from five to nine minutes. The simulation time of the provided solution methodology using evolutionary algorithms can be reduced, since evolutionary algorithms are suitable for distributed computing. The minimum pressure requirement in this case could not be satisfied and p_{min} had to be reduced to 4 (m) in order for the ACD scheme to achieve a solution. In addition, the pressure control input had to reach the upper bound of its constraints to achieve this minimum pressure in the network, which is why the median head increase is 40 (m) in all cases. This is again due to the looped nature of this network and it may could be avoided with additional

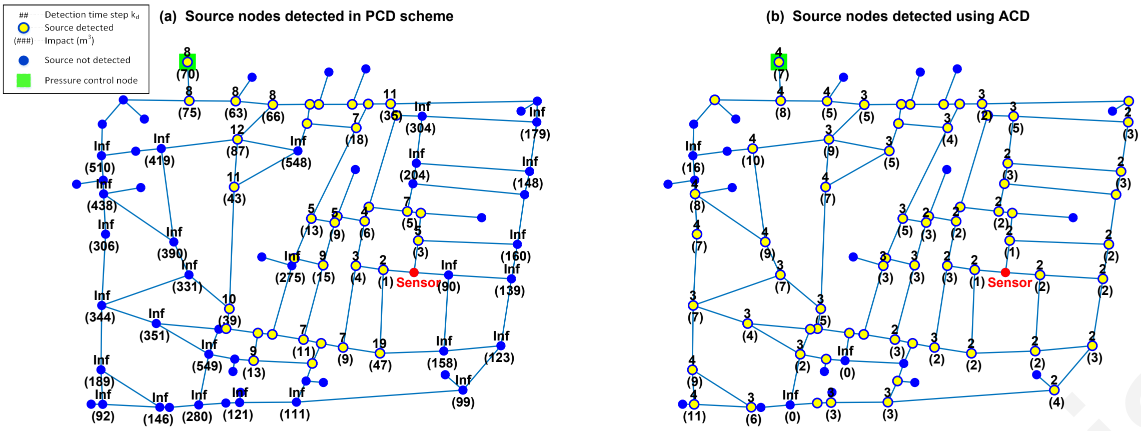

## Detection time step k_d
Source detected
(###) Impact (m^3)
Source not detected
Pressure control node
(a) Source nodes detected in PCD scheme
(b) Source nodes detected using ACD
Sensor

Figure 9.9. CY-DMA network with 1 optimally placed sensor: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

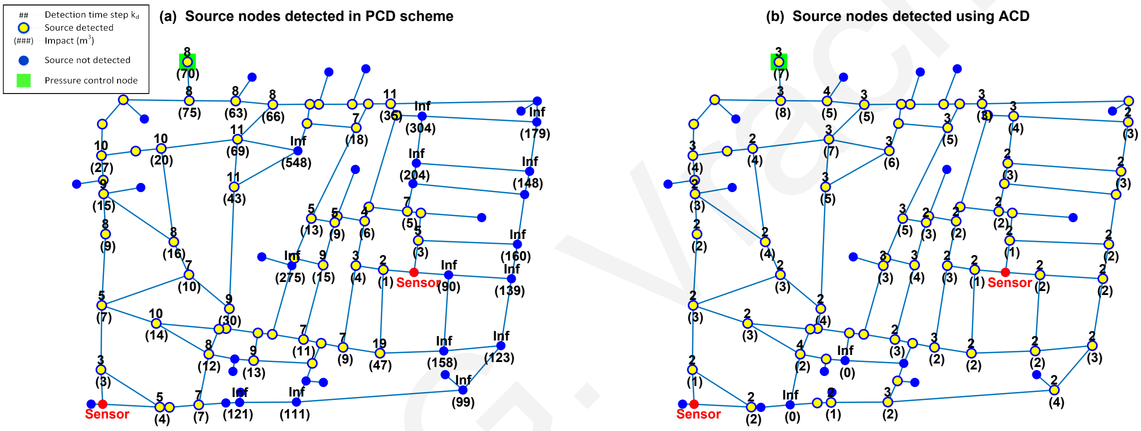

## Detection time step k_d
Source detected
(###) Impact (m^3)
Source not detected
Pressure control node
(a) Source nodes detected in PCD scheme
(b) Source nodes detected using ACD
Sensor

Figure 9.10. CY-DMA network with 2 optimally placed sensors: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

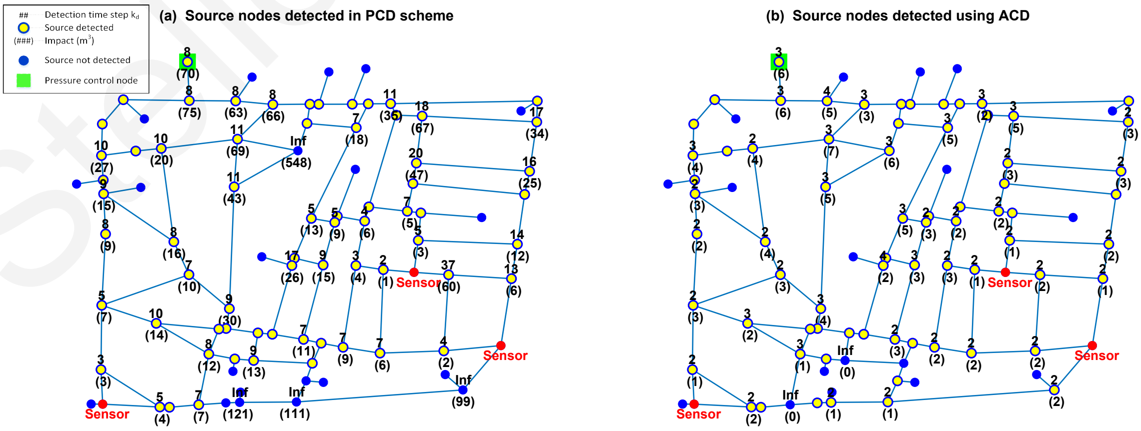

## Detection time step k_d
Source detected
(###) Impact (m^3)
Source not detected
Pressure control node
(a) Source nodes detected in PCD scheme
(b) Source nodes detected using ACD
Sensor

Figure 9.11. CY-DMA network with 3 optimally placed sensors: Detection time (impact) calculated for all possible contaminant source nodes (a) in the default PCD scheme, (b) when using the ACD scheme.

pressure controlled nodes. System resilience can be affected, which is why this methodology should be used in emergency situations.

	Cov (%)		$\tilde{k}_d$ (hours)		$\tilde{I}$ (m^3)		$\tilde{VC}$	$\tilde{H}+$ (m)
	PCD	ACD	PCD	ACD	PCD	ACD	ACD	ACD
M1								
1 Sensor	57.5	57.5	4.5	1.5	1939	392	7	40
2 Sensors	72.5	72.5	1.5	1	256	114	5	40
3 Sensors	90	90	1	1	85	65	5.5	40

Cov: Percentage of nodes monitored by sensors.

$\tilde{k}_d$: Median contamination detection time in hours.

$\tilde{I}$: Median contamination impact in m^3.

$\tilde{VC}$: Median number of valves closed by the ACD scheme.

$\tilde{H}+$: Median head increase at pressure control nodes in meters.

Table 9.2. Result metrics from simulation scenarios using Passive (PCD) and Active (ACD) contamination detection schemes on M1 networks with 1 to 3 sensors.

9.7.1 Discussion of Results

In this section the results of “Transport Network example” and “District Metered Area example” case studies are discussed and evaluated. Specific metrics are defined and given in Table 9.3 to help in this process. The following metrics are defined:

- Cov: Network coverage, defined as the percentage of contamination source nodes monitored by the installed sensors in all scenarios.
- $\tilde{k}_d$: The median of detection times in the scenarios when a source node is classified as being monitored.
- $\tilde{I}$: The median contamination impact in all scenarios, defined as the contaminated water consumed.
- $\tilde{VC}$: The median number of valves closed in the scenarios when a source node is classified as being monitored when using the ACD methodology.
- $\tilde{H}$: The median head at the pressure control node in the scenarios when a source node is classified as being monitored when using the ACD methodology.

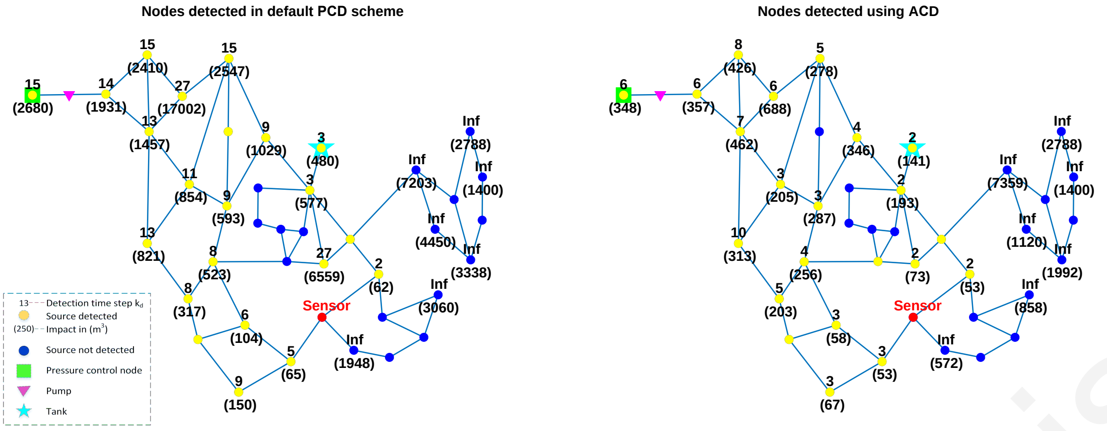


Figure 9.12. M1 network with 1 optimally placed sensor. Left: Contaminant nodes and detection time (impact) for default PCD case. Right: Contaminant nodes and detection time (impact) with ACD scheme.

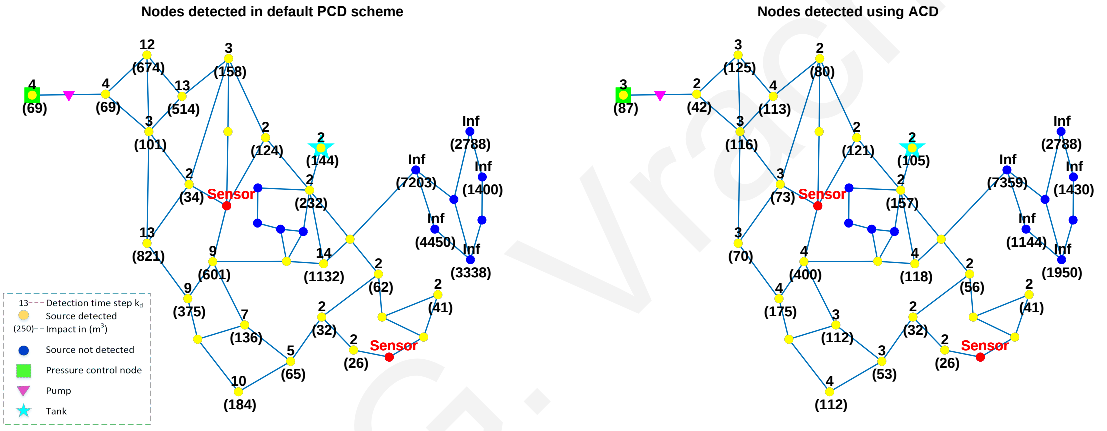


Figure 9.13. M1 network with 2 optimally placed sensors. Left: Contaminant nodes and detection time (impact) for default PCD case. Right: Contaminant nodes and detection time (impact) with ACD scheme.

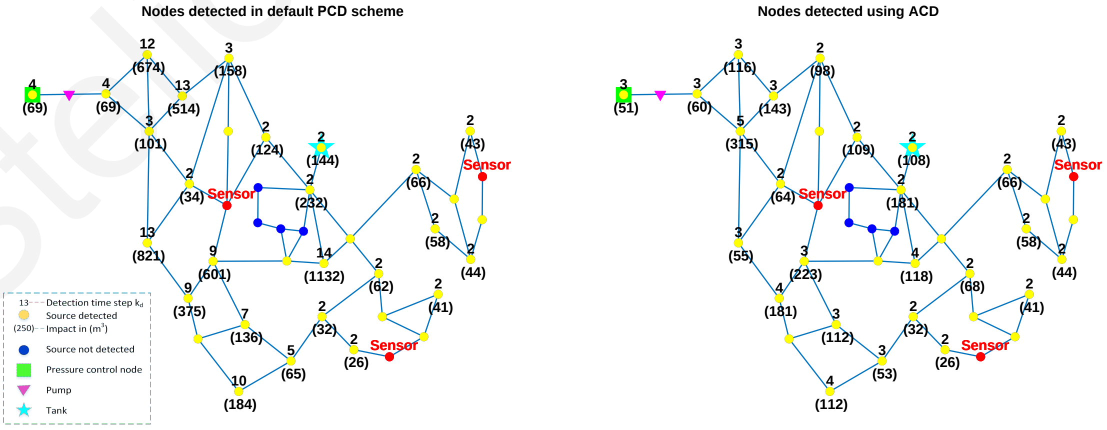


Figure 9.14. M1 network with 3 optimally placed sensors. Left: Contaminant nodes and detection time (impact) for default PCD case. Right: Contaminant nodes and detection time (impact) with ACD scheme.

Networks & Sensors	Cov (%)		$\tilde{k}_d$ (hours)		$\tilde{I}$ (m^3)		$\tilde{VC}$	$\tilde{H}$ (m)
	PCD	ACD	PCD	ACD	PCD	ACD	ACD	ACD
Hanoi								
1 Sensor	40.6	75	6.5	2.5	11398	1642	2	120
2 Sensors	65.6	78.1	5.5	2.5	4962	1289	2	105
3 Sensors	68.8	81.3	4	2	3807	956	2	115
CY-DMA								
1 Sensor	40.7	74.7	4	1.5	91	3	8	40
2 Sensors	61.5	79.1	4	1	24	3	6	40
3 Sensors	76.9	80.2	4	1	16	2	6	40

Cov:	Percentage of nodes monitored by sensors.
$\tilde{k}_d$:	Median contamination detection time in hours.
$\tilde{I}$:	Median contamination impact in m^3.
$\tilde{VC}$:	Median number of valves closed by the ACD scheme.
$\tilde{H}$:	Median head at pressure control nodes in meters.

Table 9.3. Result metrics from simulation scenarios using Passive (PCD) and Active (ACD) contamination detection schemes on Hanoi and CY-DMA networks with 1 to 3 sensors.

The results on “Transport Network example” indicate a significant increase of network coverage, especially in the case of one sensor, as seen in Table 9.3. A notable outcome is that network coverage when using the ACD scheme with one sensor (Fig. 9.6(b)) is better than the case of three optimally placed sensors in the PCD scheme (Fig. 9.8(a)). This implies that if a water utility has information about a contamination *a priori*, e.g., from customer complaints, it is possible to confirm the existence of a contaminant using fewer sensors. Note that the sensors in this case study were optimally placed for the PCD scheme, meaning that the ACD performance can still be improved with a dedicated sensor placement methodology. In terms of simulation time, each simulation needs approximately 3 minutes to run on a personal computer with Intel Core i5-2400 CPU at 3.10GHz.

In general, a significant reduction of contamination impact is observed when the ACD scheme is used, as observed by the median Impact in Table 9.3. Even when both methodologies detect the contamination, significant decrease in impact is observed as seen from Figures 9.6 – 9.8. Note that the median impact is reduced by approximately ten times. Reduced impact is also accompanied by reduced detection time. All these benefits come at the cost of valve actions and increased pressure in the network, even though the hydraulic constraints are satisfied. The median number of control inputs for the Hanoi network are two valve actions per simulation, while the median pressure set-point ranges from 105 to 120(m) depending on the scenario.

The results on “District Metered Area example” again indicate an increase in coverage, especially in the case of 1 and 2 sensors, as seen in Table 9.3. The contamination impact when using the ACD scheme is reduced by approximately a factor of eight, similar to the transport network of “Hanoi”. Valve actions are increased, with the median number of valve actions ranging from six to eight, due to this being a larger network in terms of nodes. The pressure set-points are significantly lower in this example as it is a network with residential consumers. The distribution of pressure set-points and the number of valve actions for each network-sensor case are graphically shown in Fig. 9.15 and respectively.

9.8 Sensitivity analysis

In order to assess the sensitivity of the algorithm to the detection threshold c_{thr}, additional simulations are performed on the “Hanoi” and “CY-DMA” networks, for the case where two sensors are installed, and this threshold is varied between the values of 4%, 7% and 10%. The results from these simulations indicate that the coverage is not affected by the change in

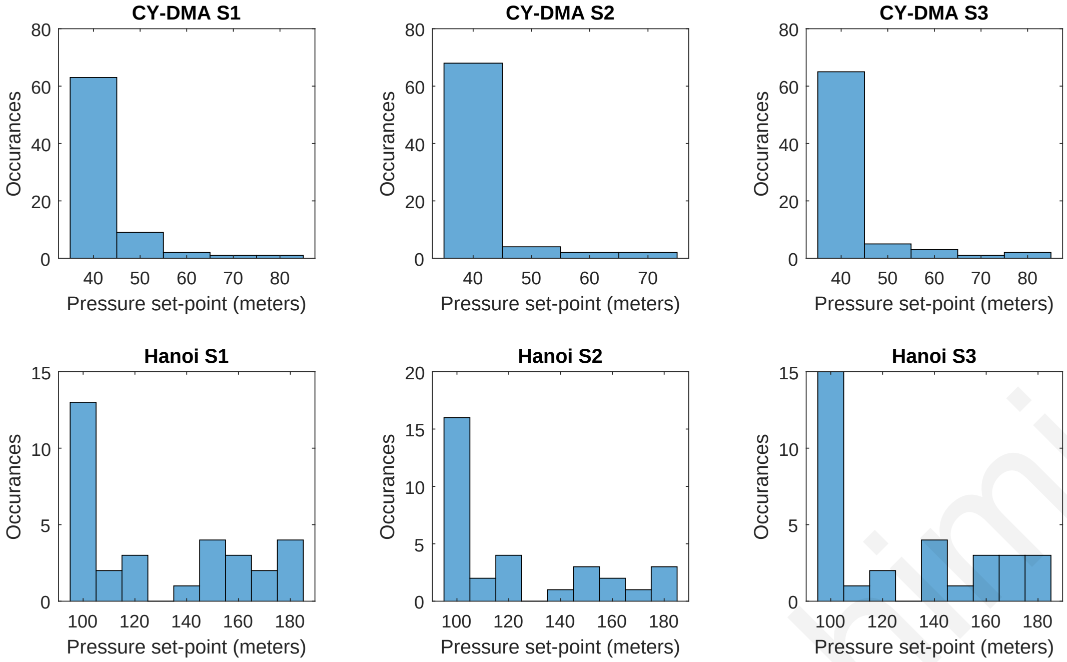


Figure 9.15. Distribution of pressure set-points for each scenario

threshold, while the impact is increased only marginally when the threshold increases.

In order to assess the performance of ACD methodology when there is uncertainty in water demands and model of the network, sensitivity analysis has been performed. Specifically, for each solution calculated by the ACD algorithm for a given contamination scenario (i.e., for a given network, source and sensor nodes), 1000 additional Monte Carlo Simulations (MCS) were performed where the demands and roughness coefficients of pipes were randomly varied between $\pm 2, 4, 6, 8$ and 10% of their estimated value, following a uniform distribution. Aggregated metrics for each uncertainty case are given in Table 9.4. The metrics used are: 1) The percentage of simulations where monitoring of the contaminant was achieved using the control settings calculated by the ACD algorithm. 2) The Mean Percentage Deviation of the contamination Impact mean calculated in the MCS, related to the Impact calculated by the ACD algorithm, defined as follows:

$$I_{MPMD} = \frac{1}{n_n} \sum_{i=1}^{n_n} 100 \left(I_i^{ACD} - \mu\left(I_i^{MCS}\right) / I_i^{ACD} \right) \tag{9.11}$$

where $\mu\left(I_i^{MCS}\right)$ is the mean Impact from the MCS for contamination scenario i and I_i^{ACD} is the Impact calculated by the ACD algorithm for scenario i. 3) The Mean Percentage Standard Deviation of the Impact calculated by the MCS for each scenario, related to the mean Impact of each scenario, defined as:

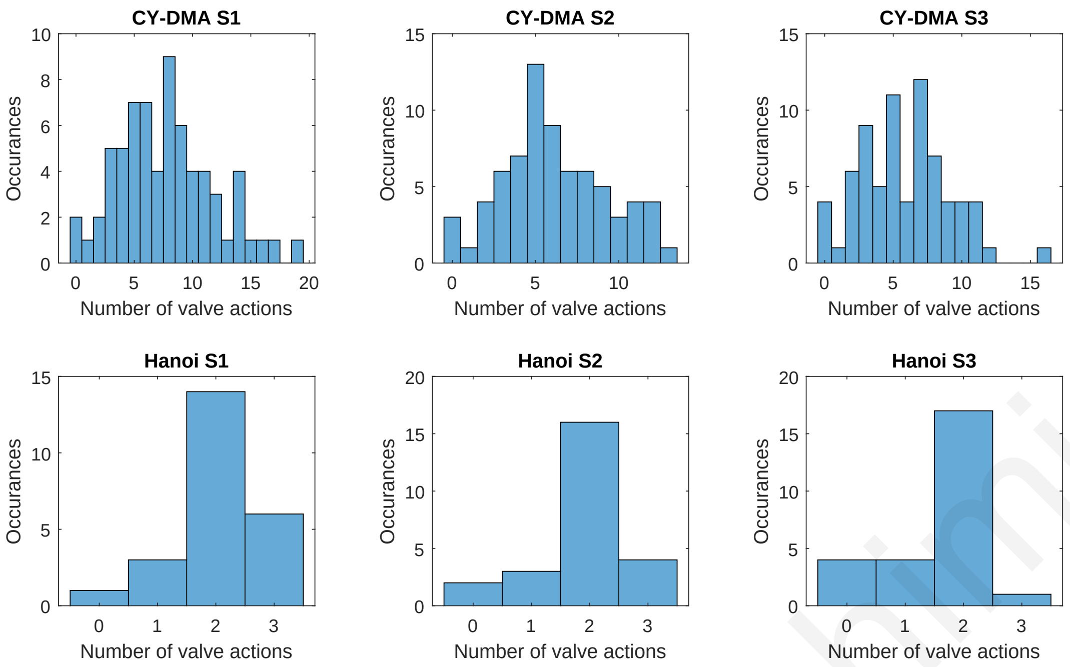


Figure 9.16. Distribution of number of valve actions for each scenario

$$I_{MPSD} = \frac{1}{n_n} \sum_{i=1}^{n_n} 100 \left(\sigma \left(I_i^{MCS} \right) / \mu \left(I_i^{MCS} \right) \right) \tag{9.12}$$

where $\sigma\left(I_i^{MCS}\right)$ is the standard deviation of Impact from the MCS of scenario i.

The results show that *the existence* of the solution provided by the ACD algorithm is robust to demand and model uncertainty, as the suspected nodes are monitored in at least 99.6% of the varied scenarios. The variation of the contamination impact under these uncertainties, as shown in Table 9.4, increases proportionally to the increase of uncertainty. It is observed that the Impact variation may be network dependent, as it is of smaller magnitude in the Hanoi network which has less nodes compared to the CY-DMA network.

9.9 Conclusions and Future Work

In this work we address the problem of water contamination detection in Water Distribution Networks (WDN) by applying a methodology of Active Contamination Detection (ACD), which manipulates the system inputs in a way that assists the contamination detection process. A generalized formulation of this problem is provided. Hydraulic constraints are incorporated into the problem, thus providing solutions that maintain the system in operation. A maximum detection time limit takes into account the scenario when manual sampling is preferable to be performed. Due to the complexity of this problem, a simplified version of

	Hanoi			CY-DMA		
Uncertainty (%)	Solved (%)	I-MPMD (%)	I-MPSD (%)	Solved (%)	I-MPSD (%)	I-MPSD (%)
2	100.00	3.95	5.35	100.00	11.41	13.10
4	100.00	6.22	7.61	100.00	17.83	18.80
6	100.00	7.65	8.51	100.00	22.23	21.79
8	99.92	9.18	8.83	100.00	26.22	23.48
10	99.61	9.68	9.19	100.00	29.73	24.40

Uncertainty:	Variation of combined water demand and roughness in the MCS as a percentage of their known values
Solved:	Percentage of MCS where monitoring of the contaminant was achieved using the settings calculated by the ACD algorithm
I-MPMD	Mean Percentage Deviation of the mean Impact from the MCS from the Impact calculated by the ACD algorithm
I-MPSD	Mean Percentage Standard Deviation of the Impact from the MCS

Table 9.4. Sensitivity analysis results of the ACD methodology applied to the Hanoi and CY-DMA networks

this problem is solved using evolutionary algorithms. Simulations are performed on networks where sensors are optimally placed for Passive Contamination Detection (PCD). The results show that it is possible to significantly reduce contamination impact, which in this work is defined as the contaminated water consumed, by manipulating the system valves and pressure set-points. Moreover, the system coverage is improved and monitoring time is decreased.

The intended use of the proposed methodology is to provide an additional tool at the disposal of water managers that can be used in conjunction with conventional water quality monitoring. A water utility may have a few specialized contamination sensors within a system (which may be of high cost), in parallel to a larger number of general water quality sensors (e.g., chlorine, ORP etc). A suspicion of a contamination may then arise due to abnormal readings of a general water quality sensor and confirmed by driving the contamination to a specialized sensor using the ACD methodology. When using general water quality sensors to identify abnormalities in water quality, measurements from more than one sensor and knowledge of water flows in the network can be combined using a suitable contamination detection methodology [176] to pinpoint a suspected contamination location. This location may not necessarily have a sensor installed.

Another example use of the proposed methodology is the case of a sensor fault which has as a result a substantial reduction in coverage. The sensor-fault can be obvious, such as the sensor giving a constant zero reading, thus the water utility will know not to rely on the this sensor. The ACD methodology can then be used to provide additional coverage with less sensors, as seen from the simulation results in this work. The impact of the contamination when using ACD is lower than when using PCD with the same number of sensors because the methodology tries to minimize this metric.

The ACD methodology does not require operable valves at all the pipes of the network, as it is able to work given a set of existent valves in the network. The calculated solution depends on the number of valves, location of valves and topology of the network. Current water networks may lack the Industrial Control System (ICS) infrastructure to remotely control valves and apply timely emergency response. However, the development trend of modern water distribution systems is towards remote sensing and control. If a water utility wants to employ such a system specifically for the use of the ACD methodology, optimal valve placement oriented towards increasing the efficiency of ACD can significantly reduce the valves needed and provide sufficient coverage.

When using the proposed methodology, there is a trade-off between increased monitoring capability and system resilience. A way to mitigate this problem is to include in the optimization procedure a metric of system resilience and take that into account when choosing a solution. Additionally, this methodology requires the knowledge of a set of nodes which are suspect of contamination in order to be applied. Sensitivity analysis with respect to water demand and model parameter variations, show that the existence of a solution to the ACD problem is robust to these uncertainties, whereas the impact value is more sensitive. In future work, these uncertainties can be taken into account in the Problem Formulation in order for the methodology to provide a more robust solution. This work could also benefit from a study which assesses the performance of this methodology on networks with different characteristics, because the existence of a solution that does not violate pressure constraints greatly depends on the networks topology. Other extensions of this work include further investigation of the weight β in the optimization cost function, modifications to redirect flows to flushing locations when the presence of a contaminant is confirmed and modification to focus on creating isolated paths between locations suspect of contamination and sensor nodes, in order to improve the localization of a contamination source.

9.10 Data Availability

The data, models, or code generated or used during the study are available in a repository or online. [https://doi.org/10.5281/zenodo.2566001]

Chapter 10

Conclusions and Future Work

10.1 Conclusions

The modernization of WDS with the installation of sensors and computer systems to process sensor data offers the opportunity for real-time monitoring and fault-diagnosis in these critical infrastructures. Due to the small number of measurements compared to the system size and the significant modeling uncertainty, *a priori* available information should be used to give a useful estimation of the system state. This thesis offers methodologies for state-estimation and fault-diagnosis which address these practical difficulties. They are designed to work in real-time using acquired measurements to deduce the state and/or give a decision about the existence of a fault.

The practical implementation of the developed model-based techniques requires an integrated platform for water distribution systems management which can host these algorithms in the form of plug-in software modules. The platform would enable real-time implementation by handling the information exchange between operators, software modules, sensors and actuators of the system. Moreover, it would be responsible to visually deliver the output information of algorithms to the operators, in a way that facilitates the decision making process. Such a platform would also handle the integration of system information needed by the model-based methodologies of this thesis, such as:

- GIS containing network spatial data
- Hydraulic and water-quality model, including bounds on parameter estimates
- Historical billing data
- Population densities and building information from open data
- Sensor specifications and locations

Additionally, some pre-processing software modules are required before applying the proposed real-time algorithms, such as a data validation module to pre-process sensor readings and remove outliers or ‘bad’ readings caused by sensor faults, and a demand pseudo-measurement generator which is able to give not only estimates of the demands but also bounds on these estimates based on the available historical information.

The key assumption of this thesis is that uncertain quantities in WDS are better described by error bounds on their estimates; in comparison, traditional state-estimation methodologies assume a known statistical characterization of the uncertainty, especially on measurements. We make the case that due to the use of pseudo-measurements this is an unrealistic assumption for water networks, and as a result, the main contribution of this thesis is the development of methodologies for solving hydraulic and water-quality models containing bounded uncertainties. Through the use of these methodologies, network operators can gain a complete view of the system, while also having a quantified measure of state estimation uncertainty.

Specifically for hydraulic-state estimation, a novel methodology is proposed that generates interval-state estimates (IHISE), by modeling uncertainties as intervals taking into account the water demand uncertainty and modeling uncertainty in the form of uncertain pipe parameters. The results show that the proposed methodology is able to generate tight bounds on hydraulic states and can be used in place of randomized methods such as Monte-Carlo Simulations (MCS). The advantage of this methodology over MCS is that the analytically calculated bounds guarantee the inclusion of the true system state, while the iterative nature of the algorithm makes these bounds as tight as possible. Moreover, the IHISE algorithm requires less computational resources compared to the large number of MCS required to obtain comparable results. However, similarly to MCS, the computational complexity increases with the number of states in the examined network.

The availability of a hydraulic-state estimation algorithm also enables water-quality state estimation due to the interconnected nature of hydraulic and water-quality dynamics. Water flows are modeled as an input to water-quality models and since interval-flow estimates are available, the development of a methodology which is able to use them is proposed. Modeling of water-quality dynamics contains significant uncertainty in the reaction rates of the substances of interest, which are affected not only by the chemical characteristics of the provided treated water but also by the physical characteristics of pipes in the network. Reaction rate uncertainty is also modeled by a bounded estimates. The proposed Backtracking Uncertainty Bounding Algorithm (BUBA) is able to calculate tight chlorine concentration bounds, given bounded hydraulic-state estimates and bounds on water quality parameters.

The bounds are validated and compared to a large number of MCS. An often ignored characteristic of water-quality models is that they contain time-varying parameters. The BUBA is designed to facilitate time-varying bulk reaction rates, which occur in systems with multiple water sources. This is demonstrated by designing a real-time parameter estimation algorithm which estimates the bulk reaction coefficients in tanks with water originating from different sources. The parameter estimation algorithm and the proposed BUBA are then applied on a real case study and an improvement is shown in the estimated chlorine concentration bounds.

The proposed state-estimation methodologies can be naturally used with model-based fault-diagnosis methodologies, because many of these rely on the availability of bounding state estimates which are calculated by some knowledge of the system uncertainties. Specifically we address the problem of leakages in WDS which is of great interest to water utilities due to the lost revenue from un-billed water but also due to the environmental impact of wasting this increasingly scarce resource. A methodology for leakage detection in WDS is proposed by directly using hydraulic interval-state estimates derived by the IHISE algorithm as detection thresholds. The methodology is applied on a case study of a modified transport network in Cyprus, using real data, with the aim to determine the existence of unaccounted-for water. The results show that the existence of an artificial leakage into the data can be detected, due to state bound violation. The practice of evaluating leakage detection and diagnosis algorithms on real network is followed by many researchers, however for them to evaluate and compare their fault diagnosis methodologies on the same case studies is difficult, as these may be considered as confidential. This is why in this thesis we develop a widely accessible dataset of real or realistic leakage scenarios, the Leakage Diagnosis Benchmark (LeakDB), which can be used as a common benchmark for research reproducibility. The dataset contains multiple scenarios and networks under varying conditions, to provide an objective assessment of the fault diagnosis algorithms, while also providing specific metrics for the evaluation of the algorithms with which we evaluate a number of detection algorithms.

The evaluation of the leakage detection methodology using directly interval-state estimates on the LeakDB demonstrated low detectability rates. A new methodology for leakage detection and localization in WDS is proposed using an model invalidation. An interval-model is formulated using historical data on demands and by deriving bounds on the uncertainty from this information. Demand calibration data in the form of group demand estimates are also incorporated into the model. Leakage detection is performed by incorporating real-time measurements from the network and checking the validity of the healthy interval-model. The results show increased detectability rates compared to the previous approach.

Moreover, a methodology for leakage localization is proposed, based on this interval-model, which yields a priority node-list that seems to significantly reduce the leakage search space for water utilities.

Another critical fault in WDS is water contamination, which we address in this thesis by proposing a model-based contamination event detection algorithm using multiple detection thresholds, which monitor multiple chlorine concentration measurements in real-time. In the first version of this algorithm, MCS are executed in parallel to the real system operation, while at each time-step, based on the known input and by assuming known bounds on the demands and roughness coefficients, an estimate of the chlorine concentration lower-bound is calculated. However, the MCS generated bounds cannot guarantee the inclusion of chlorine measurements due to the complex uncertain dynamics of the system, resulting in many false positives. Instead, a detection delay threshold can be estimated, based on historical or simulated data. However, when considering the lower-bound, this threshold can be large in order to reduce false positives. To reduce false negatives due to the use of conservative bounds, a multi-level threshold approach is adopted in this work. In specific, a set of pre-defined levels are computed as percentages of the lower-bound estimate, and for each level a detection delay threshold is calculated based on a training dataset describing normal and faulty conditions. The results demonstrate the ability of the algorithm to identify a number of contamination events with a very small number of false positives and with significant reduction in the detection delay. Moreover, we show how to generate multiple level thresholds using the analytical BUBA methodology in combination with hydraulic interval-state estimation, which in contrast to MCS it is able to guarantee the inclusion of the actual concentration in the contamination-free case given the provided water-quality model. Additionally, this approach is able to utilize available measurements from the network to improve the generated thresholds.

Even with a very good contamination detection scheme, the scarcity of sensors may not guarantee the detection of all possible contaminants in the network. Moreover, the uses of general-purpose sensors, such as chlorine sensors, only give an indication of the existence of a contaminant and it does not confirm it. A water utility may have a few specialized contamination sensors within a system (which may be of high cost), in parallel to a larger number of general water quality sensors (e.g., free chlorine, ORP etc). To address the above problems, this thesis proposes a novel methodology for altering the area monitored by water quality sensors in Water Distribution Systems (WDS) when there is suspicion of a contamination event. The proposed Active Contamination Detection (ACD), manipulates the system inputs in a way that assists the contamination detection process. Hydraulic constraints are incorporated into the problem, thus providing solutions that maintain the system in operation. A maximum detection time limit takes into account the scenario when manual sampling is preferable to be performed. Due to the complexity of this problem, a simplified version of this problem is solved using evolutionary algorithms. The results show that it is possible to significantly reduce contamination impact, which in this work is defined as the contaminated water consumed, by manipulating the system valves and pressure set-points. Moreover, the system coverage is improved and monitoring time is decreased. The intended use of the proposed methodology is to provide an additional tool at the disposal of water managers that can be used in conjunction with conventional water quality monitoring.

10.2 Future work

The main assumption of this thesis is that bounds on the uncertain parameters and measurements are known; we will refer to this as *input uncertainty*. The availability of input uncertainty bounds is detrimental to the derivation of bounds on state estimates using the proposed methodologies, which we will refer to here as *output uncertainty*. The derived output uncertainty bounds may be conservative depending on how conservative the input uncertainty bounds are. In order for the proposed state estimation methodologies to be "useful" for network operators, the input uncertainty must be relatively low. Future work will aim to quantify the aforementioned sentence, i.e., define when do the bounded state estimates become uncertain enough to not assist the operator in taking decisions on specific tasks, and what is the minimum input uncertainty which will have this result. Following on the previous point, another area of research which will greatly benefit this thesis is developing methodologies for defining the bounds on input uncertainty. Below we list some of these areas of research.

The determination of bounds on water demands at nodes when using pseudo-measurements should be investigated. As mentioned in this thesis, there exists a number of sources of information to determine an estimate of demands at nodes: information from billing data, from combining typical consumer patterns, population densities and building data, and from Automatic Meter Readings. Moreover, the effect of grouping consumers together at nodes should be investigated as this may lead in loss of accuracy in the demand pseudo-measurement, or it may be beneficial in the sense that the demand uncertainty of a group of consumers may be described more accurately by a normal distribution. Parameter uncertainty bounds such as roughness coefficients can also be defined using properties of pipes such as material and age. The most straightforward form of input uncertainty is sensor measurement noise, which can be quantified by the provided sensor specifications. Sensors specifications define the maximum error that a sensor can give, or provide a statistical characterization of the measurement error. In cases when the error is characterized by a normal distribution, the 2σ or 3σ limits can be used as measurement bounds, covering 95% or 99.7% of the possible measured state values respectively.

The availability of statistical characterization of some uncertainties in WDS raises the question of how this can be used in conjunction with the proposed approaches for bounded-state estimation. If the 3σ limits are used as bounds, some information is lost which is something that cannot be afforded in WDS state-estimation. Future work will consider the combination of uniform probability distributions, essentially the representation of uncertainty by known bounds, with other distributions such as the normal probability distribution. This is a challenging task considering the non-linear nature of hydraulic and water-quality models.

Another consideration for future work is the validity of input uncertainty bounds on fault diagnosis. Wrongly defined bounds on input uncertainty can trigger false alarms in the proposed leakage and contamination detection methodologies. The multi-level approach proposed for contamination detection is a way to address this issue by introducing probabilities into the output uncertainty estimation. However, in order for this approach to yield meaningful probabilities for the output uncertainty this should be combined with a suitable methodology of characterizing input uncertainty and how this is propagated to the output. The output uncertainty bounds will then be represented by bounds corresponding to different levels of probability of a fault. Moreover, information from multiple time steps should be incorporated into the detection logic, examples of which have been presented in this thesis for leakage diagnosis in Chapter 6 and contamination detection in Chapter 8.

The state bounds calculated by the proposed methodologies were predominantly compared with bounds computed by computationally intensive Monte-Carlo simulations. The advantage of the proposed analytical methodologies for calculating state bounds is the guarantee that the provided bounds include the true system state given the validity of the given uncertain models and inputs. However, it would be interesting to investigate and compare the proposed methodologies with more advanced methods for MCS which do not use pure random search, as it was the case for the MCS used in this work. More sophisticated methods use low-discrepancy sequences (LDS) which are sequences of points with relatively small discrepancy and are therefore distributed almost equally in the domain of interest. In this way new sampled points use information about the position of their predecessors to fill the gaps between points sampled on previous iterations. The most important advantage of LDS (e.g., Halton and Sobol sequences) is the guarantee to converge to the optimal value as more points are used [177].

The bounded state-estimation approaches presented in this work can be modified to be applied in other research fields. Specifically, the interval-state estimation approach can be applied for the state estimation of transport and distribution network of power systems. The structure and mathematical representation of power networks closely resembles that of water systems, e.g. considering the equivalence of water flow to power flow and of head-loss in pipes to power-loss in wires. Even though interval-state estimation is not a new concept in power systems literature [178, 179], a modified version of the proposed IHISE algorithm would offer the benefit of allowing the inclusion of non-linear dynamics in power systems equations, such as more accurate nonlinear representation of power-losses. Moreover, a more accurate representation of pseudo-measurements, mostly used in the absence of smart meters in distribution grids, can consider the voltage and frequency dependence of loads in the network.

The methodology for hydraulic interval-state estimation can be enhanced and improved by following the research directions listed below:

- The use of interval analysis for the calculation of state bounds is not able to capture the interdependencies between state variables. Preliminary results from the use of constrained zonotopes (CZs) which consider state-variable dependencies during the set-bounded state-estimation process has been submitted for peer review [6].
- In terms of computational complexity, this methodology was not designed to optimize execution times since these were small enough for the real-time application of this algorithm in WDS, which typically have relatively large discrete time-steps, i.e., a few minutes. The IHISE algorithm execution times can be improved significantly in a number of ways, e.g., by intelligently selecting the initial bounds to reduce the algorithm iterations, by using parallel execution of functions and by using application-specific hardware.
- Additional improvement on the computational efficiency of the proposed interval-based methodology can be achieved using different WDS equation formulations as the basis for model development. Specifically, an alternative to the *pipe formulation* used in this work is the *loop formulation*, which considers a set of independent loops with an associated flow to formulate the hydraulic equations [180]. This formulation

has the advantage of reducing the number of equations leading to a smaller system matrix, and could possibly improve the performance of the Linear Program used by the IHISE. It also has some disadvantages, such as a less sparse system matrix which is greatly affected by the selection of the independent loops [181].

- The IHISE algorithm is iterative in nature and relies on the convergence of the calculated state bounds. The linear program formulation solved at each step is ensured to have the same or smaller feasible set than the one solved in previous steps, using set operations. The reduction of the feasible set is achieved using bounding linearization and due to the inclusion isotonic nature of the nonlinear functions included in the hydraulic equations. A rigorous proof of this concept will be provided in future work.
- The current formulation does not model elements whose head function is depended on pressure, such as pressure reduction valves. In future work we will expand the formulation and the solution methodology to consider such elements. Other elements that are used in WDS and are not modeled in this work, are pressure control valves, flow control valves etc. Future work will model a variety of additional components to be used with this methodology, and an interval hydraulic state estimation toolkit will be released. Additionally, more extensive simulations on how this methodology deals with pressure-driven demands and pressure-dependent leakages will be provided.

The methodology for bounded chlorine concentration estimation arguably uses a simple model for chlorine decay dynamics. In future work we will demonstrate the applicability of this methodology using a more complex model of chlorine decay, while using the same transport model used by the BUBA. Specifically, a model using two second order terms to describe chlorine decay can be used [182], which has been shown to better approximate the reaction of chlorine in raw water. Moreover, the reactions of multiple species in water can be modeled in the same equations [98]. The additional terms may represent contaminants and can be “activated” when a detection occurs to estimate the contaminant source location.

In this thesis we have also demonstrated a real-time parameter estimation algorithm for the decay rates in tanks, however the proposed methodology is able to facilitate the use of the path information provided by the BUBA to design a wall reaction coefficient learning algorithm. This will further reduce parameter uncertainty. The quantification of the learning uncertainty in order to properly define parameter uncertainty bounds during the learning process should also be investigated.

The Active Contamination Detection (ACD) methodology can be considered the only proposed methodology of this thesis which is in early stage in terms of application readiness.

ACD requires remotely controlled valves in the network, and current water networks may lack the Industrial Control System (ICS) infrastructure for this purpose and in general the ability to apply remote and timely emergency response. The development trend of modern water distribution systems is towards remote sensing and control. If a water utility wants to employ such a system specifically for the use of the ACD methodology, optimal valve placement oriented towards increasing the efficiency of ACD can significantly reduce the valves needed and provide sufficient coverage. When using the proposed methodology, there is a trade-off between increased monitoring capability and system resilience. A way to mitigate this problem is to include in the optimization procedure a metric of system resilience and take that into account when choosing a solution. Additionally, this methodology requires the knowledge of a set of nodes which are suspect of contamination in order to be applied. Sensitivity analysis with respect to water demand and model parameter variations, show that the existence of a solution to the ACD problem is robust to these uncertainties, whereas the impact value is more sensitive. In future work, these uncertainties can be taken into account in the Problem Formulation in order for the methodology to provide a more robust solution. Other extensions of the ACD methodology include modifications to redirect flows to flushing locations when the presence of a contaminant is confirmed and modification to focus on creating isolated paths between locations suspect of contamination and sensor nodes, in order to improve the localization of a contamination source.

The proposed methodologies will benefit from investigating their application on additional real-life water distribution systems, and evaluating their effectiveness in real operational conditions. Their application through an online platform for WDS management is a challenging, but achievable, technological task which can lead to significant improvement of the way our water infrastructures are monitored and operated. The spirit of our times urges the use of technology and artificial intelligence to protect the most precious of earth's resources, water.

Bibliography

[1] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, "Real-time hydraulic interval state estimation for water transport networks: a case study," *Drinking Water Engineering and Science*, vol. 11, no. 1, pp. 19–24, 2018.

[2] S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, "Iterative hydraulic interval state estimation for water distribution networks," *Journal of Water Resources Planning and Management*, vol. 145, no. 1, p. 04018087, 2019.

[3] S. G. Vrachimis, R. Lifshitz, D. G. Eliades, M. M. Polycarpou, and A. Ostfeld, "Active contamination detection in water-distribution systems," *Journal of Water Resources Planning and Management*, vol. 146, no. 4, p. 04020014, 2020.

[4] S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, "Leakage detection and localization in water distribution systems using interval model invalidation," 2020 (submitted for review).

[5] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, "Calculating chlorine concentration bounds in water distribution networks using real-time parameter estimation," 2020 (submitted for review).

[6] B. S. Rego, S. G. Vrachimis, M. M. Polycarpou, G. V. Raffo, and D. M. Raimondo, "State estimation and leakage detection of water distribution networks using constrained zonotopes," 2020 (submitted for review).

[7] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, "Enhanced adaptive control of water quality in water distribution networks by incorporating abrupt hydraulic changes," in *Proc. of Water Distribution System Analysis WDSA2014*, vol. 89, Bari, Italy, 2014, pp. 239–246.

[8] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, "The backtracking uncertainty bounding algorithm for chlorine sensor fault detection," in *Proc. of Computing and Control for the Water Industry CCWI2015*, vol. 119, no. 1, Leicester, UK, 2015, pp. 613–622.

[9] D. G. Eliades, D. Stavrou, S. G. Vrachimis, C. G. Panayiotou, and M. M. Polycarpou, "Contamination event detection using multi-level thresholds," in *Proc. of Computing and Control for the Water Industry CCWI2015*, vol. 119, no. 1, Leicester, UK, 2015, pp. 1429–1438.

[10] D. G. Eliades, M. Kyriakou, S. G. Vrachimis, and M. M. Polycarpou, "EPANET-MATLAB Toolkit : An Open-Source Software for Interfacing EPANET with MATLAB," in *Proc. of Computing and Control for the Water Industry CCWI 2016*, Amsterdam, The Netherlands, 2016.

[11] S. G. Vrachimis, M. S. Kyriakou, D. G. Eliades, and M. M. Polycarpou, “LeakDB : A benchmark dataset for leakage diagnosis in water distribution networks,” in *WDSA / CCWI Joint Conference Proceedings*, vol. 1, Kingston, Canada, 2018.

[12] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Leak detection in water distribution systems using hydraulic interval state estimation,” in *Proc. of 2018 IEEE Conference on Control Technology and Applications (CCTA)*, Copenhagen, Denmark, 2018, pp. 565–570.

[13] S. G. Vrachimis, S. Timotheou, D. G. Eliades, and M. M. Polycarpou, “Interval state estimation of hydraulics in water distribution networks,” in *Proc. of European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 2641–2646.

[14] R. Lifshitz, A. Ostfeld, S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Improving contamination detectability in water distribution systems using active fault detection,” in *WDSA / CCWI Joint Conference Proceedings*, vol. 1, Kingston, Canada, 2018.

[15] S. G. Vrachimis, D. G. Eliades, and M. M. Polycarpou, “Calculation of chlorine concentration bounding estimates in water distribution networks using real-time learning,” in *Proc. of Computing and Control for the Water Industry CCWI 2019*, Exeter, UK, 2019.

[16] J. W. Webster, G. A. Brook, L. B. Railsback, H. Cheng, R. L. Edwards, C. Alexander, and P. P. Reeder, “Stalagmite evidence from Belize indicating significant droughts at the time of Preclassic Abandonment, the Maya Hiatus, and the Classic Maya collapse,” *Palaeogeography, Palaeoclimatology, Palaeoecology*, vol. 250, no. 1-4, pp. 1–17, jun 2007.

[17] B. Kingdom, R. Liemberger, and P. Marin, “The challenge of reducing Non-Revenue Water (NRW) in developing countries how the private sector can help: A look at performance-based service contracting,” Tech. Rep., 2006.

[18] UN, “64/292. The human right to water and sanitation,” *General Assembly*, vol. 64, no. 292, p. 3, 2010.

[19] WHO, *Guidelines for drinking-water quality, 4th edition, incorporating the 1st addendum*. WHO, 2017.

[20] W. J. Cooper, “Responding to crisis: The west virginia chemical spill,” *Environmental Science & Technology*, vol. 48, no. 6, pp. 3095–3095, 2014.

[21] European Commission, “Directive (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information,” *EUR-Lex*, vol. PE/28/2019, 2019.

[22] R. Perez, G. Sanz, V. Puig, J. Quevedo, M. A. Cuguero Escofet, F. Nejjari, J. Meseguer, G. Cembrano, J. M. Mirats Tur, and R. Sarrate, “Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in barcelona,” *IEEE Control Systems*, vol. 34, no. 4, pp. 24–36, 2014.

[23] S. Díaz, R. Mínguez, and J. González, “Topological observability analysis in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 143, no. 5, p. 06017001, 2017.

[24] A. Preis, A. J. Whittle, M. Asce, A. Ostfeld, and L. Perelman, “An efficient hydraulic state estimation technique using reduced models of urban water networks,” *Journal of Water Resources Planning and Management*, vol. 137, pp. 343–351, 2011.

[25] D. Mouly, E. Joulin, C. Rosin, P. Beaudeau, A. Zeghnoun, A. Olszewski-Ortar, J. F. Munoz, B. Welté, M. Joyeux, R. Seux, A. Montiel, and M. J. Rodriguez, “Variations in trihalomethane levels in three French water distribution systems and the development of a predictive model,” *Water Research*, vol. 44, no. 18, pp. 5168–5179, 2010.

[26] C. Y. Peng, G. V. Korshin, R. L. Valentine, A. S. Hill, M. J. Friedman, and S. H. Reiber, “Characterization of elemental and structural composition of corrosion scales and deposits formed in drinking water distribution systems,” *Water Research*, vol. 44, no. 15, pp. 4570–4580, 2010.

[27] European Commission, “Proposal for a directive of the european parliament and of the council on the quality of water intended for human consumption (recast),” p. COM/2017/0753 final 2017/0332(COD), 2018.

[28] D. E. Helbling and J. M. VanBriesen, “Modeling residual chlorine response to a microbial contamination event in drinking water distribution systems,” *Journal of Environmental Engineering*, vol. 135, no. 10, pp. 918–927, 2009.

[29] J. Hall, A. D. Zaffiro, R. B. Marx, P. C. Kefauver, R. E. Krishman, R. C. Haught, J. G. Herrmann, E. R. Krishnan, R. C. Haught, J. G. Herrmann, E. Radha Krishnan, R. C. Haught, and J. G. Herrmann, “On-line water quality parameters as indicators of distribution system contamination,” *Journal of American Water Works Association*, vol. 99, no. 1, pp. 66–77, 2007.

[30] J. C. Powell, N. B. Hallam, J. R. West, C. F. Forster, and J. Simms, “Factors which control bulk chlorine decay rates,” *Water Research*, vol. 34, no. 1, pp. 117–126, 2000.

[31] D. G. Eliades, M. Kyriakou, and M. M. Polycarpou, “Sensor placement in water distribution systems using the S-PLACE Toolkit,” in *Proc. of 12th International Conference on Computing and Control for the Water Industry, CCWI2013*, vol. 70, no. 2010, 2014, pp. 602–611.

[32] N. Islam, R. Sadiq, and M. J. Rodriguez, “Optimizing locations for chlorine booster stations in small water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 143, no. 7, p. 04017021, 2017.

[33] I. Okeya, Z. Kapelan, C. Hutton, and D. Naga, “Online burst detection in a water distribution system using the kalman filter and hydraulic modelling,” in *Proc. of 16th Water Distribution System Analysis Conference, WDSA2014*, vol. 89. Bari, Italy: Elsevier, 2014, pp. 418–427.

[34] J. Machell, S. R. Mounce, and J. B. Boxall, “Online modelling of water distribution systems: A UK case study,” *Drinking Water Engineering and Science*, vol. 3, no. 1, pp. 21–27, 2010.

[35] Z. Rao and E. Salomons, “Development of a real-time, near-optimal control process for water-distribution networks,” *Journal of Hydroinformatics*, vol. 9, no. 1, p. 25, 2007.

[36] A. Bargiela, “An algorithm for observability determination in water-system state estimation,” in *IEE Proceedings D - Control Theory and Applications*, vol. 132, no. 6, 1985, pp. 245–250.

[37] A. K. Nagar and R. S. Powell, “Observability analysis of water distribution systems under parametric and measurement uncertainty,” in *Joint Conference on Water Resource Engineering and Water Resources Planning and Management 2000: Building Partnerships*, vol. 104, no. 1. Reston, VA: American Society of Civil Engineers, 2000, pp. 1–10.

[38] S. Díaz, J. González, and R. Mínguez, “Uncertainty evaluation for constrained state estimation in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 142, no. 12, p. 06016004, 2016.

[39] R. S. Powell, M. R. Irving, M. J. H. Sterling, and A. Usman, “A comparison of three real-time state estimation methods for on-line monitoring of water distribution systems,” *Computer applications in water supply*, vol. 1, pp. 333–348, 1988.

[40] J. H. Andersen, R. S. Powell, and J. F. Marsh, “Constrained state estimation with applications in water distribution network monitoring,” *International Journal of Systems Science*, vol. 32, no. 6, pp. 807–816, 2001.

[41] D. Kang and K. Lansey, “Real-time demand estimation and confidence limit analysis for water distribution systems,” *Journal of Hydraulic Engineering*, vol. 135, no. 10, pp. 825–837, 2009.

[42] C. J. Hutton, Z. Kapelan, L. Vamvakeridou-Lyroudia, and D. A. Savić, “Dealing with uncertainty in water distribution system models: A framework for real-time modeling and data assimilation,” *Journal of Water Resources Planning and Management*, vol. 140, no. 2, pp. 169–183, 2014.

[43] N. Avni, B. Fishbain, and U. Shamir, “Water consumption patterns as a basis for water demand modeling,” *Water Resources Research*, vol. 51, no. 10, pp. 8165–8181, 2015.

[44] A. Bargiela and G. D. Hainsworth, “Pressure and flow uncertainty in water systems,” *Journal of Water Resources Planning and Management*, vol. 115, no. 2, pp. 212–229, 1989.

[45] Z. S. Kapelan, D. A. Savic, and G. A. Walters, “Multiobjective sampling design for water distribution model calibration,” *Journal of Water Resources Planning and Management*, vol. 129, no. 6, pp. 466–479, 2003.

[46] D. Kun, L. Tian-Yu, W. Jun-Hui, and G. Jin-Song, “Inversion model of water distribution systems for nodal demand calibration,” *Journal of Water Resources Planning and Management*, vol. 141, no. 9, p. 04015002, 2015.

[47] S. Díaz, J. González, and R. Mínguez, “Observability analysis in water transport networks: Algebraic approach,” *Journal of Water Resources Planning and Management*, vol. 142, no. 4, p. 04015071, 2016.

[48] D. A. Savic, Z. S. Kapelan, and P. M. Jonkergouw, “Quo vadis water distribution model calibration?” *Urban Water Journal*, vol. 6, no. 1, pp. 3–22, 2009.

[49] W. Cheng and Z. He, “Calibration of nodal demand in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 137, no. 1, pp. 31–40, 2011.

[50] G. Sanz, R. Pérez, Z. Kapelan, and D. Savic, “Leak detection and localization through demand components calibration,” *Journal of Water Resources Planning and Management*, vol. 142, no. 2, p. 04015057, 2016.

[51] N. C. Do, A. R. Simpson, J. W. Deuerlein, and O. Piller, “Calibration of water demand multipliers in water distribution systems using genetic algorithms,” *Journal of Water Resources Planning and Management*, vol. 142, no. 11, p. 04016044, 2016.

[52] S. Díaz, R. Mínguez, J. González, and D. Savic, “Explicit expressions for state estimation sensitivity analysis in water systems,” *Journal of Water Resources Planning and Management*, vol. 144, no. 4, 2018.

[53] T. Gao, “Pipe roughness estimation in water distribution networks using head loss adjustment,” *Journal of Water Resources Planning and Management*, vol. 143, no. 5, p. 04017007, 2017.

[54] A. Bargiela, W. Pedrycz, and M. Tanaka, “A study of uncertain state estimation,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 3, pp. 288–301, 2003.

[55] R. Łangowski and M. Brdys, “Monitoring of chlorine concentration in drinking water distribution systems using an interval estimator,” *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 2, pp. 199–216, 2007.

[56] R. Pérez, V. Puig, J. Pascual, A. Peralta, E. Landeros, and L. Jordanas, “Pressure sensor distribution for leak detection in Barcelona water distribution network,” *Water Science & Technology: Water Supply*, vol. 9, no. 6, p. 715, 2009.

[57] M. Brdys and K. Chen, “Joint state and parameter estimation of dynamic water supply system under bounded uncertainty using geometric programming,” *IFAC Proceedings Volumes*, vol. 27, no. 8, pp. 1287–1292, 1994.

[58] B. Gabrys and A. Bargiela, “Integrated neural based system for state estimation and confidence limit analysis in water networks,” in *Proc. of The 8th European Simulation Symposium, Ess 96. Society for Computer Simulation*. Society for Computer Simulation, 1997, pp. 398–402.

[59] C. T. Arsene, D. Al-Dabass, and J. Hartley, “Confidence limit analysis of water distribution systems based on a least squares loop flows state estimation technique,” in *Proc. of 2011 UKSim 5th European Symposium on Computer Modeling and Simulation*. IEEE, 2011, pp. 94–101.

[60] M. F. K. Pasha and K. Lansey, “Effect of parameter uncertainty on water quality predictions in distribution systems-case study,” *Journal of Hydroinformatics*, vol. 12, no. 1, pp. 1–21, 2010.

[61] T. K. Chan, C. S. Chin, and X. Zhong, “Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection,” *IEEE Access*, vol. 6, pp. 78 846–78 867, 2018.

[62] S. G. Vrachimis, D. G. Eliades, R. Taormina, A. Ostfeld, Z. Kapelan, S. Liu, M. S. Kyriakou, P. Pavlou, M. Qiu, and M. M. Polycarpou, "BattLeDIM 2020 - Battle of the Leakage Detection and Isolation Methods."

[63] R. Li, H. Huang, K. Xin, and T. Tao, "A review of methods for burst/leakage detection and location in water distribution systems," *Water Science and Technology: Water Supply*, vol. 15, no. 3, pp. 429–441, 2015.

[64] R. Puust, Z. Kapelan, D. A. Savic, and T. Koppel, "A review of methods for leakage management in pipe networks," *Urban Water Journal*, vol. 7, no. 1, pp. 25–45, 2010.

[65] D. Zaman, M. K. Tiwari, A. K. Gupta, and D. Sen, "A review of leakage detection strategies for pressurised pipeline in steady-state," *Engineering Failure Analysis*, p. 104264, 2019.

[66] D. Misiunas, J. Vítkovský, G. Olsson, A. Simpson, and M. Lambert, "Pipeline break detection using pressure transient monitoring," *Journal of Water Resources Planning and Management*, vol. 131, no. 4, pp. 316–325, 2005.

[67] A. F. Colombo, P. Lee, and B. W. Karney, "A selective literature review of transient-based leak detection methods," *Journal of Hydro-environment Research*, vol. 2, no. 4, pp. 212–227, 2009.

[68] J. P. Vítkovský, A. R. Simpson, and M. F. Lambert, "Leak detection and calibration using transients and genetic algorithms," *Journal of Water Resources Planning and Management*, vol. 126, no. 4, pp. 262–265, 2000.

[69] S. Mounce, A. Day, A. Wood, A. Khan, P. Widdop, and J. Machell, "A neural network approach to burst detection," *Water Science and Technology*, vol. 45, no. 4-5, pp. 237–246, 2002.

[70] S. R. Mounce, J. B. Boxall, and J. Machell, "Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows," *Journal of Water Resources Planning and Management*, vol. 136, no. 3, pp. 309–318, 2010.

[71] D. G. Eliades and M. M. Polycarpou, "Leakage fault detection in district metered areas of water distribution systems," *Journal of Hydroinformatics*, vol. 14, no. 4, pp. 992–1005, 2012.

[72] A. Soldevila, J. Blesa, S. Tornil-Sin, E. Duviella, R. M. Fernandez-Canti, and V. Puig, "Leak localization in water distribution networks using a mixed model-based/data-driven approach," *Control Engineering Practice*, vol. 55, pp. 162–173, 2016.

[73] Y. Wu and S. Liu, "A review of data-driven approaches for burst detection in water distribution systems," *Urban Water Journal*, vol. 14, no. 9, pp. 972–983, 2017.

[74] L. A. Rossman, *EPANET 2: users manual*, 2000.

[75] R. Pérez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta, "Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks," *Control Engineering Practice*, vol. 19, no. 10, pp. 1157–1167, 2011.

[76] B. Farley, S. R. Mounce, and J. B. Boxall, "Field testing of an optimal sensor placement methodology for event detection in an urban water distribution network," *Urban Water Journal*, vol. 7, no. 6, pp. 345–356, 2010.

[77] M. Casillas, V. Puig, L. Garza-Castañón, and A. Rosich, "Optimal sensor placement for leak location in water distribution networks using genetic algorithms," *Sensors*, vol. 13, no. 11, pp. 14 984–15 005, 2013.

[78] M. À. Cugueró-Escofet, V. Puig, and J. Quevedo, "Optimal pressure sensor placement and assessment for leak location using a relaxed isolation index: Application to the Barcelona water network," *Control Engineering Practice*, vol. 63, pp. 1–12, 2017.

[79] R. S. Pudar and J. A. Liggett, "Leaks in pipe networks," *Journal of Hydraulic Engineering*, vol. 118, no. 7, pp. 1031–1046, 1992.

[80] J.-A. Goulet, S. Coutu, and I. F. Smith, "Model falsification diagnosis and sensor placement for leak detection in pressurized pipe networks," *Advanced Engineering Informatics*, vol. 27, no. 2, pp. 261–269, 2013.

[81] J. Moors, L. Scholten, J. P. van der Hoek, and J. den Besten, "Automated leak localization performance without detailed demand distribution data," *Urban Water Journal*, vol. 15, no. 2, pp. 116–123, 2018.

[82] Z. Y. Wu, P. Sage, and D. Turtle, "Pressure-dependent leak detection model and its application to a district water system," *Journal of Water Resources Planning and Management*, vol. 136, no. 1, pp. 116–128, 2010.

[83] L. Ribeiro, J. Sousa, A. Marques, and N. Simões, "Locating leaks with trustrank algorithm support," *Water*, vol. 7, no. 4, pp. 1378–1401, 2015.

[84] D. Kang and K. Lansey, "Demand and roughness estimation in water distribution systems," *Journal of Water Resources Planning and Management*, vol. 137, no. 1, pp. 20–30, 2011.

[85] P. Rosa, C. Silvestre, J. S. Shamma, and M. Athans, "Fault detection and isolation of LTV systems using set-valued observers," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 768–773.

[86] J. Blesa, V. Puig, J. Saludes, and J. Vento, "Leak detection, isolation and estimation in pressurized water pipe networks using lpv models and zonotopes," *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 36–41, 2010.

[87] P. Subramaniam, M. E. Tryby, and J. G. Uber, "Set covering models for locating booster chlorination stations in water distribution systems." *Joint Conference on Water Resource Engineering and Water Resources Planning and Management 2000: Building Partnerships*, no. 1, pp. 1–9, 2000.

[88] W. E. Hart and R. Murray, "Review of sensor placement strategies for contamination warning systems in drinking water distribution systems," *Journal of Water Resources Planning and Management*, vol. 136, no. 6, pp. 611–619, 2010.

[89] D. G. Eliades and M. M. Polycarpou, "A fault diagnosis and security framework for water systems," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1254–1265, 2010.

[90] D. Zeng, L. Gu, L. Lian, S. Guo, H. Yao, and J. Hu, "On cost-efficient sensor placement for contaminant detection in water distribution systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2177–2185, 2016.

[91] L. A. Rossman and P. F. Boulos, "Numerical methods for modeling water quality in distribution systems: A comparison," *Journal of Water Resources Planning and Management*, vol. 122, no. 2, pp. 137–146, 1996.

[92] M. M. Polycarpou, J. G. J. G. Uber, Z. W. Z. Wang, F. S. F. Shang, and M. Brdys, "Feedback control of water quality," *IEEE Control Systems*, vol. 22, no. 3, pp. 68–87, 2002.

[93] K. S. Tshehla, Y. Hamam, and A. M. Abu-Mahfouz, "State estimation in water distribution network: A review," in *Proc. of 15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 1247–1252.

[94] F. Hua, J. R. West, R. A. Barker, and C. F. Forster, "Modelling of chlorine decay in municipal water supplies," *Water Research*, vol. 33, no. 12, pp. 2735–2746, 1999.

[95] R. M. Clark, W. M. Grayman, J. A. Goodrich, R. A. Deininger, and K. Skov, "Measuring and modeling chlorine propagation in water distribution systems," *Journal of Water Resources Planning and Management*, vol. 120, no. 6, pp. 871–887, 1994.

[96] L. Monteiro, D. Figueiredo, D. Covas, and J. Menaia, "Integrating water temperature in chlorine decay modelling: a case study," *Urban Water Journal*, vol. 14, no. 10, pp. 1097–1101, 2017.

[97] R. M. Clark, "Chlorine Demand and TTHM Formation Kinetics: A Second-Order Model," *Journal of Environmental Engineering*, vol. 124, no. 1, pp. 16–24, 1998.

[98] F. Shang, J. G. Uber, and L. A. Rossman, "Modeling reaction and transport of multiple species in water distribution systems," *Environmental Science & Technology*, vol. 42, no. 3, pp. 808–814, 2008.

[99] L. A. Rossman, R. M. Clark, and W. M. Grayman, "Modeling chlorine residuals in drinking-water distribution systems," *Journal of Environmental Engineering*, vol. 120, no. 4, pp. 803–820, 1994.

[100] J. J. Vasconcelos, L. A. Rossman, W. M. Grayman, P. F. Boulos, and R. M. Clark, "Kinetics of chlorine decay," *Journal of American Water Works Association*, vol. 89, no. 7, pp. 54–65, 1997.

[101] F. Shang, J. G. Uber, and M. M. Polycarpou, "Particle backtracking algorithm for water distribution system analysis," *Journal of Environmental Engineering*, vol. 128, no. 5, pp. 441–450, 2002.

[102] M. Zierolf, M. Polycarpou, and J. Uber, "Development and autocalibration of an input-output model of chlorine transport in drinking water distribution systems," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 4, pp. 543–553, 1998.

[103] R. Isermann, *Fault-Diagnosis Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[104] T. Raïssi, D. Efimov, and A. Zolghadri, "Interval state estimation for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 260–265, 2012.

[105] B. S. Rego, G. V. Raffo, J. K. Scott, and D. M. Raimondo, "Guaranteed methods based on constrained zonotopes for set-valued state estimation of nonlinear discrete-time systems," *Automatica*, vol. 111, p. 108614, 2020.

[106] R. Łangowski and M. A. Brdys, "An interval estimator for chlorine monitoring in drinking water distribution systems under uncertain system dynamics, inputs and chlorine concentration measurement errors," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 2, pp. 309–322, 2017.

[107] T. P. Lambrou, C. G. Panayiotou, and C. C. Anastasiou, "A low-cost system for real time monitoring and assessment of potable water quality at consumer sites," in *2012 IEEE Sensors*. IEEE, 2012, pp. 1–4.

[108] A. Ostfeld, J. G. Über, E. Salomons, J. W. J. W. Berry, W. E. W. E. Hart, C. A. C. a. Phillips, J.-P. J.-P. J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, F. di Pierro, S.-T. S.-T. Khu, D. Savic, D. Eliades, M. Polycarpou, S. R. S. R. Ghimire, B. D. B. D. Barkdoll, R. Gueli, J. J. J. J. Huang, E. a. E. A. McBean, W. James, A. Krause, J. Leskovec, S. Isovitsch, J. Xu, C. Guestrin, J. VanBriesen, M. Small, P. Fischbeck, A. Preis, M. Propato, O. Piller, G. B. G. B. Trachtman, Z. Yi Wu, T. Walski, J. G. Uber, E. Salomons, J. W. J. W. Berry, W. E. W. E. Hart, C. A. C. a. Phillips, J.-P. J.-P. J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, F. di Pierro, S.-T. S.-T. Khu, D. Savic, D. Eliades, M. Polycarpou, S. R. S. R. Ghimire, B. D. B. D. Barkdoll, R. Gueli, J. J. J. J. Huang, E. a. E. A. McBean, W. James, A. Krause, J. Leskovec, S. Isovitsch, J. Xu, C. Guestrin, J. VanBriesen, M. Small, P. Fischbeck, A. Preis, M. Propato, O. Piller, G. B. G. B. Trachtman, Z. Y. Wu, and T. Walski, "The Battle of the Water Sensor Networks (BWSN): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, p. 556, 2008.

[109] Y. Zhao, R. Schwartz, E. Salomons, A. Ostfeld, and H. V. Poor, "New formulation and optimization methods for water sensor placement," *Environmental Modelling & Software*, vol. 76, pp. 128–136, 2016.

[110] S. Campbell, K. Horton, and R. Nikoukhah, "Auxiliary signal design for rapid multi-model identification using optimization," *Automatica*, vol. 38, no. 8, pp. 1313–1325, 2002.

[111] C. S. C. Hood and C. Ji, "Proactive network-fault detection," *IEEE Transactions on Reliability*, vol. 46, no. 3, pp. 333–341, 1997.

[112] S. L. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004, vol. 11.

[113] D. Byer and K. H. Carlson, "Expanded summary: Real-time detection of intentional chemical contamination in the distribution system," *Journal of American Water Works Association*, no. 7, pp. 130–133, 2005.

[114] R. Jarrett, G. Robinson, and R. O'Halloran, "On-line monitoring of water distribution systems: Data processing and anomaly detection," in *Proc. Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, USA, 2006.

[115] D. Kroll and K. King, "Laboratory and flow loop validation and testing of the operational effectiveness of an on-line security platform for the water distribution system," in *Proc. of Water Distribution Systems Analysis Symposium 2006*. Reston, VA: American Society of Civil Engineers, 2008, pp. 1–16.

[116] D. B. Hart and S. A. McKenna, "CANARY Users Manual," Albuquerque, NM 87185-0735, 2009.

[117] R. Murray, T. Haxton, R. Janke, W. E. Hart, J. Berry, and C. Phillips, "Water quality event detection systems for drinking water contamination warning systems—development, testing, and application of CANARY," U.S. EPA, Tech. Rep., 2010.

[118] S. Murray, M. Ghazali, and E. A. McBean, "Real-time water quality monitoring: Assessment of multisensor data using bayesian belief networks," *Journal of Water Resources Planning and Management*, vol. 138, no. 1, pp. 63–70, 2011.

[119] L. Perelman, J. Arad, M. Housh, and A. Ostfeld, "Event detection in water distribution systems from multivariate water quality time series," *Environmental science & technology*, vol. 46, no. 15, pp. 8212–8219, 2012.

[120] J. Arad, M. Housh, L. Perelman, and A. Ostfeld, "A dynamic thresholds scheme for contaminant event detection in water distribution systems," *Water research*, vol. 47, no. 5, pp. 1899–1908, 2013.

[121] N. Oliker and A. Ostfeld, "Minimum volume ellipsoid classification model for contamination event detection in water distribution systems," *Environmental Modelling & Software*, vol. 57, pp. 1–12, 2014.

[122] P. M. R. Jonkergouw, S. T. Khu, and D. Savic, "Chlorine: A possible indicator of intentional chemical and biological contamination in a water distribution network," in *Proc. of IWA Conference on Automation in Water Quality Monitoring (AutMoNet), Vienna, Austria*, 2004, p. 8.

[123] K. Umberg, J. G. Uber, and R. Murray, "Performance evaluation of real-time event detection algorithms," in *Proc. of Water Distribution Systems Analysis Symposium 2006*. Reston, VA: American Society of Civil Engineers, 2008, pp. 1–14.

[124] Y. Jeffrey Yang, R. C. Haught, and J. A. Goodrich, "Real-time contaminant detection and classification in a drinking water pipe using conventional water quality sensors: Techniques and experimental results," *Journal of Environmental Management*, vol. 90, no. 8, pp. 2494–2506, 2009.

[125] D. G. Eliades, M. P. Michaelides, C. G. Panayiotou, and M. M. Polycarpou, "Security-oriented sensor placement in intelligent buildings," *Building and Environment*, vol. 63, no. null, pp. 114–121, 2013.

[126] D. G. Eliades, C. Panayiotou, and M. M. Polycarpou, "Contamination event detection in drinking water systems using a real-time learning approach," in *Proc. of 2014 International Joint Conference on Neural Networks (IJCNN)*, IEEE. IEEE, 2014, pp. 663–670.

[127] P. F. Boulos, K. E. Lansey, and B. W. Karney, *Comprehensive Water Distribution Systems Analysis Handbook for Engineers and Planners*. MWH Soft, Incorporated, 2006.

[128] E. Todini and S. Pilati, "A gradient algorithm for the analysis of pipe networks," in *Proc. of International Conference on Computer Applications for Water Supply and Distribution*, no. September. Leucester: Research Studies Press Ltd., 1988, pp. 1–20.

[129] J. M. Wagner, U. Shamir, and D. H. Marks, "Water Distribution Reliability: Simulation Methods," *Journal of Water Resources Planning and Management*, vol. 114, no. 3, pp. 276–294, 1988.

[130] O. Giustolisi and D. Laucelli, “Water distribution network pressure-driven analysis using the Enhanced Global Gradient Algorithm (EGGA),” *Journal of Water Resources Planning and Management*, vol. 137, no. 6, pp. 498–510, 2011.

[131] K. A. Klise, D. Hart, D. M. Moriarty, M. L. Bynum, R. Murray, J. Burkhardt, and T. Haxton, “Water network tool for resilience (WNTR) User manual,” Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), Tech. Rep., 2017.

[132] O. Giustolisi, D. Savic, and Z. Kapelan, “Pressure-driven demand and leakage simulation for water distribution networks,” *Journal of Hydraulic Engineering*, vol. 134, no. 5, pp. 626–635, 2008.

[133] M. Daumas, D. Lester, and C. Munoz, “Verified real number calculations: A library for interval arithmetic,” *IEEE Transactions on Computers*, vol. 58, no. 2, pp. 226–237, 2009.

[134] R. B. Kearfott, “Interval Computations: Introduction, Uses, and Resources,” *Euromath Bulletin*, vol. 2, no. 1, pp. 95–112, 1996.

[135] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*. Siam, 2009.

[136] E. J. M. Blokker, J. H. G. Vreeburg, and J. C. van Dijk, “Simulating residential water demand with a stochastic end-use model,” *Journal of Water Resources Planning and Management*, vol. 136, no. 1, pp. 19–26, 2010.

[137] S. Díaz, R. Mínguez, and J. González, “Topological state estimation in water distribution systems: Mixed-integer quadratic programming approach,” *Journal of Water Resources Planning and Management*, vol. 144, no. 7, p. 04018026, 2018.

[138] F. Pasqualetti, F. Dorfler, and F. Bullo, “Attack detection and identification in cyber-physical systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[139] V. L. Do, L. Fillatre, and I. Nikiforov, “A statistical method for detecting cyber/physical attacks on SCADA systems,” in *Proc. of IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 364–369.

[140] A. Neumaier and A. Pownuk, “Linear systems with large uncertainties, with applications to truss structures,” *Reliable Computing*, vol. 13, no. 2, pp. 149–172, 2007.

[141] L. V. Kolev, “A method for outer interval solution of parametrized systems of linear interval equations,” *Reliable Computing*, vol. 10, pp. 227–239, 2004.

[142] L. V. Kolev, “An improved interval linearization for solving nonlinear problems,” *Numerical Algorithms*, vol. 37, no. 1-4 SPEC. ISS., pp. 213–224, 2004.

[143] C. Jiang, X. Han, and G. Liu, “A sequential nonlinear interval number programming method for uncertain structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 49-50, pp. 4250–4265, 2008.

[144] E. Zieniuk, M. Kapturczak, and A. Kużelewski, “Solving interval systems of equations obtained during the numerical solution of boundary value problems,” *Computational and Applied Mathematics*, vol. 35, no. 2, pp. 629–638, 2015.

[145] J. W. Chinneck and K. Ramadan, “Linear programming with interval coefficients,” *The Journal of the Operational Research Society*, vol. 51, no. 2, pp. 209–220, 2000.

[146] G. H. Huang and M. F. Cao, “Analysis of solution methods for interval linear programming,” *Journal of Environmental Informatics*, vol. 17, no. 2, pp. 54–64, 2011.

[147] F. Mazenc, T. N. Dinh, and S. I. Niculescu, “Interval observers for discrete-time systems,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2867–2890, 2014.

[148] T. M. Walski, E. D. Brill, J. Gessler, I. C. Goulter, R. M. Jeppson, K. Lansey, H. Lee, J. C. Liebman, L. Mays, D. R. Morgan, and L. Ormsbee, “Battle of the Network Models: Epilogue,” *Journal of Water Resources Planning and Management*, vol. 113, no. 2, pp. 191–203, 1987.

[149] M. D. Jolly, A. D. Lothes, L. Sebastian Bryson, and L. Ormsbee, “Research database of water distribution system models,” *Journal of Water Resources Planning and Management*, vol. 140, no. 4, pp. 410–416, 2014.

[150] T. McCabe, “A Complexity Measure,” *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, 1976.

[151] V. Puig, “Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies,” *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 4, pp. 619–635, 2010.

[152] F. Harirchi and N. Ozay, “Guaranteed model-based fault detection in cyber–physical systems: A model invalidation approach,” *Automatica*, vol. 93, pp. 476–488, 2018.

[153] N. Ozay, M. Sznaier, and C. Lagoa, “Convex certificates for model (in)validation of switched affine systems with unknown switches,” *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2921–2932, 2014.

[154] G. O. LLC, “Gurobi optimizer reference manual,” 2019.

[155] A. Soldevila, J. Blesa, T. N. Jensen, S. Tornil-Sin, R. M. Fernandez-Canti, and V. Puig, “Leak localization method for water-distribution networks using a data-driven model and Dempster-Shafer reasoning,” *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2020.

[156] C. Blocher, F. Pecci, and I. Stoianov, “Localizing leakage hotspots in water distribution networks via the regularization of an inverse problem,” *Journal of Hydraulic Engineering*, vol. 146, no. 4, p. 04020025, 2020.

[157] A. Rajeswaran, S. Narasimhan, and S. Narasimhan, “A graph partitioning algorithm for leak detection in water distribution networks,” *Computers & Chemical Engineering*, vol. 108, pp. 11–23, 2018.

[158] A. Kyriacou, P. Demetriou, C. Panayiotou, and E. Kyriakides, “Controlled islanding solution for large-scale power systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1591–1602, 2018.

[159] A. Kyriacou, S. Timotheou, M. P. Michaelides, C. Panayiotou, and M. Polycarpou, “Partitioning of intelligent buildings for distributed contaminant detection and isolation,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 72–86, 2017.

[160] D. Kang and K. Lansey, “Real-time optimal valve operation and booster disinfection for water quality in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 136, no. 4, pp. 463–473, 2010.

[161] M. Borza, A. S. Rambely, and M. Saraj, “Linear programming for an absolute value linear fractional programming with interval coefficients in the objective function,” *AIP Conference Proceedings*, vol. 7, no. 73, pp. 3641–3653, 2013.

[162] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006.

[163] O. Fujiwara and D. B. Khang, “A two-phase decomposition method for optimal design of looped water distribution networks,” *Water Resources Research*, vol. 26, no. 4, pp. 539–549, 1990.

[164] L. A. Rossman, P. F. Boulos, and T. Altman, “Discrete volume-element method for network water-quality models,” *Journal of Water Resources Planning and Management*, vol. 119, no. 5, pp. 505–517, 1993.

[165] D. G. Eliades and M. M. Polycarpou, “Multi-objective optimization of water quality sensor placement in drinking water distribution networks,” in *Proc. of 2007 European Control Conference (ECC)*. IEEE, 2007, pp. 1626–1633.

[166] A. Kessler, A. Ostfeld, and G. Sinai, “Detecting accidental contaminations in municipal water networks,” *Journal of Water Resources Planning and Management*, vol. 124, no. 4, pp. 192–198, 1998.

[167] A. Ostfeld and E. Salomons, “Optimal layout of early warning detection stations for water distribution systems security,” *Journal of Water Resources Planning and Management*, vol. 130, no. 5, pp. 377–385, 2004.

[168] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks, B. M. Brentan, E. Campbell, G. Lima, D. Manzi, D. Ayala-Cabrera, M. Herrera, I. Montalvo, J. Izquierdo, E. Luvizotto, S. E. Chandy, A. Rasekh, Z. A. Barker, B. Campbell, M. E. Shafiee, M. Giacomoni, N. Gatsis, A. Taha, A. A. Abokifa, K. Haddad, C. S. Lo, P. Biswas, M. F. K. Pasha, B. Kc, S. L. Somasundaram, M. Housh, and Z. Ohar, “Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 144, no. 8, p. 04018048, 2018.

[169] D. G. Eliades, M. M. Polycarpou, and B. Charalambous, “A security-oriented manual quality sampling methodology for water systems,” *Water Resources Management*, vol. 25, no. 4, pp. 1219–1228, 2011.

[170] G. Dorini, P. Jonkergouw, Z. Kapelan, and D. Savic, “SLOTS: Effective algorithm for sensor placement in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 136, no. 6, pp. 620–628, 2010.

[171] H. Ung, O. Piller, D. Gilbert, and I. Mortazavi, “Accurate and optimal sensor placement for source identification of water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 143, no. 8, p. 04017032, 2017.

[172] W. M. Grayman, R. Murray, D. A. Savic, and R. Farmani, "Redesign of water distribution systems for passive containment of contamination," *Journal of American Water Works Association*, vol. 108, no. 7, pp. E381–E391, 2016.

[173] L. W. Mays, K. Lansey, and L. W. Mays, *Hydraulic Design Handbook*. McGraw-Hill Education, 1999.

[174] D. G. D. G. Eliades and M. M. M. Polycarpou, "Fault isolation and impact evaluation of water distribution network contamination," in *IFAC World Congress Proceedings*, vol. 18, no. PART 1. Elsevier, 2011, p. 6.

[175] V. Ghorbanian, B. Karney, and Y. Guo, "Pressure standards in water distribution systems: Reflection on current practice with consideration of some unresolved issues," *Journal of Water Resources Planning and Management*, vol. 142, no. 8, p. 04016023, 2016.

[176] D. G. Eliades and M. M. Polycarpou, "Water contamination impact evaluation and source-area isolation using decision trees," *Journal of Water Resources Planning and Management*, vol. 138, no. 5, pp. 562–570, 2012.

[177] S. Kucherenko and Y. Sytsko, "Application of deterministic low-discrepancy sequences in global optimization," *Computational Optimization and Applications*, vol. 30, no. 3, pp. 297–318, 2005.

[178] Z. Wang and F. Alvarado, "Interval arithmetic in power flow analysis," *[Proceedings] Conference Papers 1991 Power Industry Computer Application Conference*, vol. 7, no. 3, pp. 156–162, 1991.

[179] C. Rakpenthai, S. Uatrongjit, and S. Premrudeeprechacharn, "State estimation of power system considering network parameter uncertainty based on parametric interval linear systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 305–313, 2012.

[180] F. Alvarruiz, F. Martínez-Alzamora, and A. M. Vidal, "Improving the efficiency of the loop method for the simulation of water distribution systems," *Journal of Water Resources Planning and Management*, vol. 141, no. 10, 2015.

[181] F. Alvarruiz, F. M. Alzamora, and A. M. Vidal, "Efficient modeling of active control valves in water distribution systems using the loop method," *Journal of Water Resources Planning and Management*, vol. 144, no. 10, 2018.

[182] R. M. Clark and M. Sivaganesan, "Predicting chlorine residuals in drinking water: Second order model," *Journal of Water Resources Planning and Management*, vol. 128, no. 2, pp. 152–161, 2002.

[183] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.

[184] D. A. Savic, G. A. Walters, and M. Schwab, "Multiobjective genetic algorithms for pump scheduling in water supply," in *AISB International Workshop on Evolutionary Computing*. Springer Berlin Heidelberg, 1997, pp. 227–235.

[185] J. E. van Zyl, D. A. Savic, and G. A. Walters, “Operational optimization of water distribution systems using a hybrid genetic algorithm,” *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 160–170, 2004.

[186] C. K. Ho and S. S. Khalsa, “EPANET-BAM: Water quality modeling with incomplete mixing in pipe junctions,” in *Proc. of Water Distribution Systems Analysis 2008*. Reston, VA: American Society of Civil Engineers, apr 2009, pp. 1–11.

[187] C. Siew and T. T. Tanyimboh, “Pressure-Dependent EPANET Extension,” *Water Resources Management*, vol. 26, no. 6, pp. 1477–1498, 2012.

[188] B. J. Eck, “An R package for reading EPANET files,” *Environmental Modelling and Software*, vol. 84, pp. 149–154, oct 2016.

[189] D. B. Steffelbauer, D. Fuchs-Hanusch, D. Steffelbauer, and D. Fuchs-Hanusch, “OOPNET: An object-oriented EPANET in Python,” in *Proc. of 13th Computing and Control for the Water Industry, CCWI 2015*, vol. 119, Leicester, UK, 2015, pp. 710–718.

[190] J. Ernst and V. Zyl, “OOTEN: An Object-oriented programmers toolkit for EPANET,” Department of Civil and Urban Engineering, University of Johannesburg, Tech. Rep., 2003.

[191] M. C. Dodd, N. D. Vu, A. Ammann, R. Kissner, H. V. Pham, M. Berg, and U. von Gunten, “Kinetics and mechanistic aspects of As (III) oxidation by aqueous chlorine, chloramines, and ozone: Relevance to drinking water treatment,” *Environmental Science & Technology*, vol. 40, no. 10, pp. 3285–3292, 2006.

[192] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “A novel video dataset for change detection benchmarking,” *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4663–4679, 2014.

[193] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, “The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition,” *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[194] A. Lavin and S. Ahmad, “Evaluating real-time anomaly detection algorithms – The numenta anomaly benchmark,” in *Proc. of IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 38–44.

[195] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, “Systematic construction of anomaly detection benchmarks from real data,” in *Proc. of the ACM SIGKDD Workshop on Outlier Detection and Description - ODD ’13*. New York, New York, USA: ACM Press, 2013, pp. 16–21.

[196] H. Yamauchi and W.-y. Huang, “Alternative models for estimating the time series components of water consumption data,” *Journal of the American Water Resources Association*, vol. 13, no. 3, pp. 599–610, 1977.

[197] G. Forman and M. Scholz, “Apples-to-apples in cross-validation studies,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, p. 49, 2010.

[198] L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2020. [Online] Available: ”http://www.gurobi.com”

[199] Spiegel Online, ‘The Nokia Virus: Finnish Town Faces Fecal Foul-up” 2008, [Online] Available: ”https://www.spiegel.de/international/europe/the-nokia-virus-finnish-town-faces-fecal-foul-up-a-527057.html”

[200] UN News, “Unsafe water kills more people than war, Ban says on World Day” 2010, [Online] Available: ”https://news.un.org/en/story/2010/03/333182-unsafe-water-kills-more-people-war-ban-says-world-day”

Appendices

Appendix A΄

EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB

In this work we introduce the EPANET-MATLAB Toolkit, an open-source software for interfacing a drinking water distribution system simulation library, EPANET, with the MATLAB technical computing language. The basic functionalities of the Toolkit are presented, and in the case-study a contamination simulator based on EPANET and EPANET-MSX is described and demonstrated on a benchmark network.

A΄.1 Introduction

In 1994, the US Environmental Protection Agency (EPA) released EPANET (version 1.1), an open source software developed by Lew Rossman, for modeling water distribution systems hydraulic and quality dynamics. EPANET was designed as a research tool to better understand the dynamics of drinking water constituents, taking into account bulk flow and pipe wall reactions [99]. EPANET considers a geometric representation of the pipe network, along with a set of initial conditions (e.g. water levels in tanks), rules of how the system is operated, and uses this information to compute flows, pressures and water quality (e.g. disinfection concentrations and water age) throughout the network, for a certain period of time. EPANET was originally developed in the C programming language and simulations can be executed through a stand-alone compiled application using a text file which contained the model description. EPANET utilizes the "gradient algorithm" for solving the hydraulic state-estimation at each time step [128]. For water quality the Finite Volume Method was originally utilized [164], however, a Lagrangian approach [91] was adopted in the following release of EPANET (version 2.0). This new version allowed the dynamic linking of EPANET with external software through its shared object library. In 2015, the *Open Source EPANET Initiative* was established, comprised of various academic, industrial and other stakeholders, to manage the further development of EPANET. An updated version of EPANET (version 2.1)[1] was released 2016, and the next major release is currently under development.

EPANET has been established as the *de facto* standard tool for both the industry and academia: for water operators to simulate "what-if" scenarios, for researchers to evaluate novel algorithms for a variety of research challenges using realistic water distribution network benchmarks, and for the industry to create new products and services while benefiting from EPANET's public-domain software license. During the past 20 years, EPANET was

[1] https://github.com/OpenWaterAnalytics/EPANET

extensively used as a tool to facilitate research in topics such as network design optimization [183, 184], operational optimization [185] and sensor placement [108].

A number of extensions have been released by the water community to expand EPANET's capabilities. For instance, as EPANET (version 2.0) was not able to simulate the interaction of multiple chemical agents, the EPANET-MSX extension was developed which allowed the simulation of the reaction and transport dynamics of multiple physical/chemical/biological parameters within a distribution network [98]. Another example is EPANET-BAM, which allows incomplete mixing in pipe junctions [186], as well as EPANET-PDX for pressure-driven hydraulic state estimation [187].

From a software engineering point-of-view, EPANET has been used within procedural programs through a series of direct calls to its library. This requires the user to be aware of all the different functions offered by EPANET, as well as the sequence of function calls in order to succesfully implement a simulation cycle. Moreover, there was not a common EPANET data structure in order to share data between different function modules and applications. These challenges can be addressed by adopting an Object-Oriented Programming approach. Recently, effort was given in creating software for using EPANET through Object-Oriented Programming interfaces, in different programming languages, such as R [188] and Python [189]. A significant effort in utilizing Object-Oriented Programming to expand EPANET's capabilities was by [190], who introduced OOTEN. OOTEN is comprised of different classes with associated methods (for instance the Class which describes water pipes provides functions to return pipe parameters, such as the diameter and length).

The contribution of this work is the introduction and demonstration of operation of the EPANET-MATLAB Toolkit[2] (version 2.1), an open-source software released under the European Union Public License (EUPL), developed at the *KIOS Research Center for Intelligent Systems and Networks* of the *University of Cyprus*. The Toolkit interfaces EPANET with MATLAB®, a widely-used programming environment and allows the user to have access to all the network information through a data structure, to execute direct calls to the EPANET library, to modify and to create EPANET networks, to run multi-species simulations through EPANET-MSX, as well as to visualize the network.

This chapter is organized as follows: In Section 2, the EPANET-MATLAB Toolkit is introduced, and in Section 3, a case study is presented illustrating the use of the Toolkit. Section 4 concludes the chapter and discusses future work.

A΄.2 The EPANET-MATLAB Toolkit

A΄.2.1 Interfacing with EPANET in MATLAB

EPANET can be used in two ways: 1) as a standalone executable software, or 2) as a shared object library. As a standalone executable software, EPANET can be called through a standard shell (e.g. *Command Line* in Windows). As a shared object, e.g. Dynamic Link Library (DLL) for Windows, EPANET can be called through a programming interface by external software written in different programming languages (such as C/C++, Python, MATLAB and Visual Basic). The external software can make calls to specific EPANET functions which modify system parameters, the time series and the simulation configuration.

Programming languages such as MATLAB® are used by researchers to design and evaluate new methodologies and tools for analyzing water distribution networks. MATLAB is a high-level programming environment used for data processing and analysis. It allows the development of applications in different platforms, and has build-in a large number of so-

[2] https://github.com/OpenWaterAnalytics/EPANET-Matlab-Toolkit

phisticated applications for optimization, control, signal processing and others. MATLAB is also able to connect to external software libraries, which allows researchers to use tools and simulators developed originally in a different language, such as C or C++. There are three methods of interfacing EPANET with MATLAB:

1. The first method is to make direct calls to the EPANET library, through the build-in function of the programming tool, which requires using MATLAB's build-in methods for loading and calling library functions (i.e. using the *loadlibrary* and *calllib* functions).
2. The second method is to use "wrappers"; MATLAB methods which follow similar naming conventions as the EPANET functions, that handle the communication with the library internally[3]. This is a higher-level of interfacing with the library, however, it requires the user to design custom data structures. For each EPANET function, a corresponding MATLAB function is required, and new algorithms need to be designed using those functions.
3. The third method is to use an Object-Oriented approach, by defining a MATLAB Class, which provides a standardized way to handle the network structure, to call all functions as well as procedures using multiple functions, to simulate and in general to perform different types of analysis in the network, through the corresponding object. The EPANET-MATLAB Toolkit utilizes this approach.

A´.2.2 How to use the EPANET-MATLAB Toolkit

The Toolkit is based on a MATLAB Class, **epanet**, which is composed of the following elements: 1) properties of the input network model, 2) static properties, 3) public methods (functions) that the user can use directly in MATLAB, 4) local (internal) functions the Toolkit uses to make direct calls to EPANET/EPANET-MSX. An **epanet** object is a specific instance of the **epanet** class. To create a new **epanet** object in MATLAB for a specific network (e.g. 'Net2.inp'), the following command is used[4]:

```
G = epanet('Net2.inp') % Load network and use the EPANET library
```

The element G is an object which can be defined mathematically as the set comprised of the network topology, structural parameters and functions. This **epanet** object can be shared between different MATLAB functions.

When the object is constructed, the Toolkit reads the input file and populates more than 300 object parameters (including pipe diameters and node elevations). The Toolkit can update these parameters when there is a change in the network model. Examples on how to retrieve parameter values are provided below:

```
properties(G) % Lists all available parameters
diameters = G.LinkDiameter % Link diameters from parameters
elevations = G.NodeElevations % Node elevations from parameters
```

After the construction of the object, it is possible to call the Toolkit functions. The Toolkit provides an extensive set of methods which allow the user to retrieve data and to simulate hydraulic and quality dynamics using the EPANET libraries. Some examples are provided below:

```
methods(G) % Lists all available methods
elevations = G.getNodeElevations([2 5]) % Node elevations for Nodes 2 & 5
```

[3] https://github.com/OpenWaterAnalytics/epanet-matlab

[4] The code in this section is provided at https://github.com/eldemet/ccwi2016.

```
diameters = G.getLinkDiameter % Link diameters from library
diameters(2)=18 % Change Link 2 diameter from 14 to 18
G.setLinkDiameter(diameters) % Set new link diameter
G.getLinkDiameter(2) % Confirms that Link 2 diameter is 18
G.plot % Plots the network in a MATLAB figure
A = G.getConnectivityMatrix % Construct connectivity matrix
func_list = G.getENfunctionsImpemented % EPANET functions implemented
```

To simulate the system dynamics, such as flows/pressures and water quality, various methods have been implemented to solve and retrieve the data: 1) solve using the EPANET's shared object library and get the desired results from memory (step-wise mode); 2) solve using the EPANET's shared object library and create a Binary output file, which is then read to retrieve all the results (batch mode); solve using the EPANET's executable and create a Binary output file, which is then read to retrieve all the results (batch mode). The hydraulic and quality dynamics can be computed using the following instructions:

```
H = G.getComputedHydraulicTimeSeries % Solve hydraulics in library
Q = G.getComputedQualityTimeSeries % Solve quality dynamics in library
B = G.getBinComputedAllParameters % Solve in library, create Binary file
```

To use EPANET-MSX, the user must first load an EPANET network, and then load an MSX file which describes the chemical dynamics. The Toolkit can be used to 'get/set' different MSX variables, to solve the multi-species dynamics, to plot concentrations and to write new MSX files. An example is provided below:

```
G.loadMSXFile('net2-cl2.msx') % Load MSX file with reactions
Q_msx = G.getMSXComputedQualityNode % Compute water quality using MSX
G.plotMSXSpeciesNodeConcentration(3,1) % Plot MSX species in MATLAB
```

Finally, to unload the libraries from MATLAB memory, the user should make the following calls:

```
G.unloadMSX % Unload EPANET-MSX library
G.unload % Unload EPANET library
```

A´.3 Case Study

In this section, we illustrate through a case study some of the Toolkit's key features. In specific, we consider the case where we want to model Arsenite contamination events of various magnitudes occurring at a single location within a water distribution system. Here we consider the case of creating multiple Arsenite contamination scenarios, affecting a drinking water distribution system which uses Chlorine for disinfection. As Chlorine reacts with Arsenite, its concentration is reduced; details on the reaction dynamics can be found in [123, 191]. For this case study, we consider the benchmark 'Network 1' from the Battle of the Water Sensor Placement [108, 89], and furthermore we assume that chlorine sensors have been installed at 'optimal' locations, at the nodes '17', '83', '122', '31' and '45'. The network's demands and roughness coefficients are partially known.

Typically when implementing such as software module using low-level calls to the library, a large number of commands need to be written in order to achieve specific results, such as extracting the pipe roughness coefficients or specifying a new demand pattern. Through the use of the EPANET-MATLAB Toolkit, however, a large part of the repetitive code is already included in the Toolkit functions, and can be used directly. Moreover, the use of the Toolkit facilitates the creation of modules which can be integrated in other software. In addition to that, it is possible to write the software in such as way so that it supports distributed computing, which is useful when developing cloud services.

Below, we provide a template solution on how to design an Arsenite contamination simulator based on EPANET and EPANET-MSX using the EPANET-MATLAB Toolkit. To reproduce the case study, the complete source code is provided at https://github.com/eldemet/ccwi2016.

The EPANET Input and MSX files are loaded as follows, constructing the **epanet** object G:

```
G = epanet('BWSN_Network_1.inp'); % Load EPANET Input file
G.loadMSXFile('Arsenite.msx'); % Load MSX file
```

The locations of the chlorine sensors are declared as:

```
sensor_id = {'JUNCTION-17', 'JUNCTION-83', 'JUNCTION-122', 'JUNCTION-31',
    'JUNCTION-45'};
sensor_index = G.getNodeIndex(sensor_id);
```

The duration of the case-study is set to 5 days. The network parameters (demand patterns and roughness coefficients) are retrieved, to be used for randomizing the hydraulics parameters.

```
t_d = 5; % days
G.setTimeSimulationDuration(t_d*24*60*60); % Set simulation duration
demand_pattern = G.getPattern;
roughness_coeff = G.getLinkRoughnessCoeff;
node_id = G.getNodeNameID;
```

A scenario matrix is constructed, which is comprised of all the contamination injection parameters (location, magnitude of concentration, start-time, duration). In addition, the uncertainty in demands and roughness coefficients is defined.

```
Ns = 100; % Number of scenarios to simulate
u_p = 0.20; u_r = 0.20; % pattern/roughness uncertainty
max_inj_conc = 2.0; %maximum Arsenic source concentration
inj_start_time = 2*48; % after day 2 (Dt = 30min)
inj_duration = 24; % maximum duration of 12 hours
inj_sc=[randi(G.NodeCount,Ns,1), max_inj_conc*rand(Ns,1),
   randi(48,Ns,1)+inj_start_time, randi(inj_duration,Ns,1)]; % Scenarios
```

The main part of the algorithm, is composed of the algorithms for adding uncertainty to the demand patterns and roughness coefficients, as well as for adding contaminant sources and solving the quality dynamics.

```
for i = 1:Ns
   G.setPatternMatrix(add_unc(demand_pattern, r_p)); % Randomize patterns
   G.setLinkRoughnessCoeff(add_unc(roughness_coeff, r_r)); % roughness
   G.setMSXSources(node_id(inj_sc(i,1)), 'AsIII', 'Setpoint', inj_sc(i,2),
      'AS3PAT'); % Specify Arsenite injection source
   G.setMSXPattern('AS3PAT',create_pat(t_d, inj_sc)); % Injection pattern
   Q{i} = G.getMSXComputedQualityNode(sensor_index); % Solve dynamics
   G.setMSXSources(node_id(inj_sc(i,1)), 'AsIII', 'Setpoint', 0, 'AS3PAT');
      % Reset injection source
```

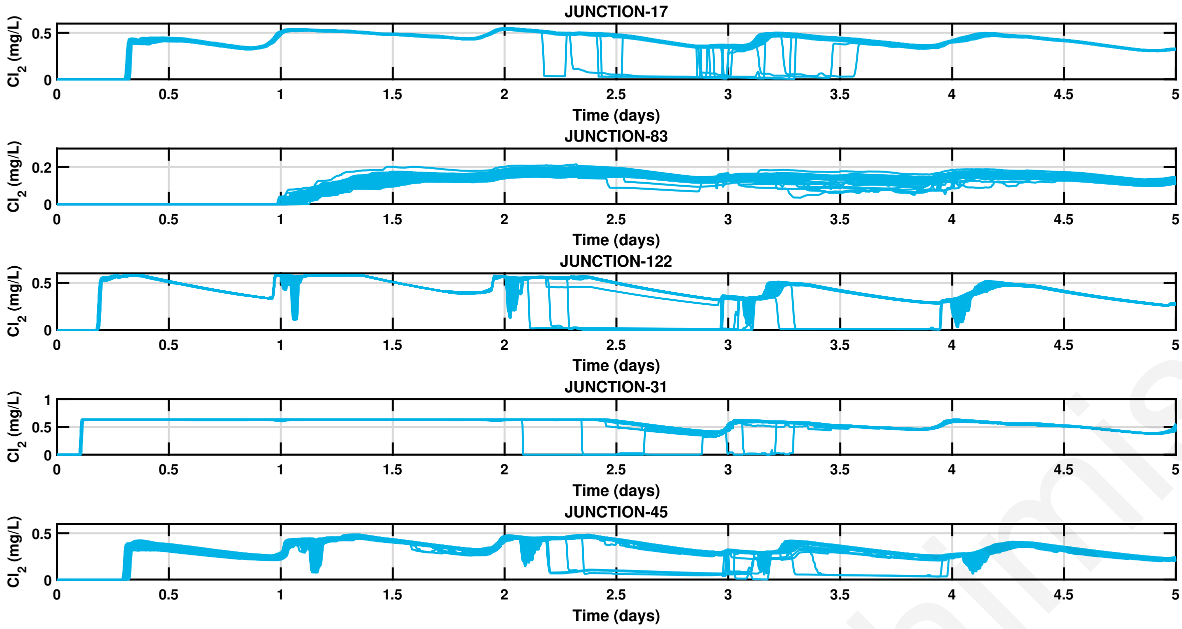


Figure A´.1. Overlay of the estimated chlorine concentration at 5 nodes from the benchmark network, from 100 random simulation scenarios.

```
end
```

The results of this case study are depicted in Fig. A´.1, in which the estimated chlorine concentrations for 100 scenarios are overlaid on the same graph. This illustrates the variability of chlorine concentrations during normal operation; the drops in concentration correspond to contamination events, after the second day.

The collection of these time-series can be used as a benchmark for evaluating the effectiveness of various Contamination Event Diagnosis algorithms *in silico*, as in [9].

A´.4 Conclusions

In this work we have presented the EPANET-MATLAB Toolkit, an open-source software for interfacing MATLAB with EPANET, in an intuitive and an easy-to-use way. The Toolkit allows the user to access EPANET and EPANET-MSX through their shared object libraries, as well as their executables. The potential uses of the Toolkit expand to a wide range of applications; for instance, the Toolkit can be used as a framework to design new monitoring, control and fault/event diagnosis algorithms, as well as to formulate optimization problems which are based on water distribution network models. The modular architecture of the Toolkit allows the user to easily expand its capabilities by adding new functions. Furthermore, as a data structure, the Toolkit can facilitate exchange of data between function modules. The EPANET-MATLAB Toolkit is released under an open-source license (EUPL), where any user can contribute, report issues or make suggestions for improvement, following the best practices of the EPANET Open Source Initiative. Future work will expand the Toolkit's capabilities, so that it is cross-platform and can be used directly in distributed algorithms which are suitable for cloud-based big-data analysis.

Appendix B΄

LeakDB: A benchmark dataset for leakage diagnosis in water distribution networks

The increase of streaming data from water utilities is enabling the development of real-time anomaly and fault detection algorithms that can detect events, such as pipe bursts and leakages. Currently, there is not a widely accessible dataset of real or realistic leakage scenarios, which could be used as a common benchmark to compare different algorithms, as well as to support research reproducibility. In this chapter we propose the design of a realistic leakage dataset, the *Leakage Diagnosis Benchmark* (LeakDB). The dataset is comprised of a large number of artificially created but realistic leakage scenarios, on different water distribution networks, under varying conditions. Additionally, a scoring algorithm was developed in MATLAB to evaluate the results of different algorithms using various metrics. The usage of the LeakDB dataset, is demonstrated by scoring four detection algorithms. The dataset is stored on an open research data repository, and will be updated in the future with new simulation scenarios. The source code of the toolkit that generates the leakage benchmark dataset, as well as the detection algorithms used, are released as open source.

B΄.1 Introduction

Developments in hydraulic sensor technologies and on-line data acquisition systems enable water companies to deploy a larger number of pressure and flow sensing devices within their systems. As a result, a large volume of streaming, time-series data is being collected, representing the state of hydraulic dynamics. These data are mainly used for regulatory reporting by water operators. In addition to reporting, these real-time data can be used for detecting events, such as pipe bursts and leakages. This is typically achieved by visual analysis and limit checking. In the literature, advanced anomaly and fault detection algorithms have been developed in the previous years, which are potentially more capable of distinguishing between actual events and false alarms, typically by comparing sensor measurements with some model. In practice, these algorithms need to be trained and evaluated on large volumes of data. However, researchers developing leakage detection and diagnosis algorithms, may have limited access to real datasets originating from industrial partners, as these may be considered as confidential. As a result, it is difficult for researchers to evaluate and compare their fault diagnosis methodologies on realistic water distribution systems.

Currently, there is not a widely accessible dataset of real or realistic leakage scenarios, which could be used as a common benchmark, as well as for research reproducibility. Such a dataset should contain multiple scenarios and networks under varying conditions, to provide an objective assessment of the fault diagnosis algorithms. Benchmarks have been extensively used in multiple applications, such as image/video processing [192], body activity recognition [193], anomaly detection in times-series [194], and many more which are available in public repositories, such as the UCI Machine Learning Repository [1] and the Penn Machine Learning Benchmark [2]. In the water distribution systems analysis community, various EPANET network benchmarks have been presented in the past [149], as well as data provided in the context of conference competitions, e.g., for solving sensor placement problems (BWSN) [108] or for detecting cyber-physical attacks (BATtle of the Attack Detection ALgorithms - BATADAL) [168].

In this chapter we propose the *Leakage Diagnosis Benchmark* (LeakDB) dataset to promote research and evaluation of leakage detection and isolation algorithms. The requirements of the benchmark are analyzed based on systematic approaches for constructing benchmark datasets, e.g., as in [195]. The benchmark is comprised of a large number of realistic leakage scenarios which occur randomly at different water distribution benchmark networks, of different size and topology. For each benchmark network and for each leakage scenario, the leakage parameters (e.g., number of leaks, locations, size), the structural parameters (e.g., length, pipe roughness) and realistic consumer pressure-driven demands are varied. The dataset is comprised of all leakage scenario parameters, hydraulic dynamics (flows, pressures), node demands and the network model. The benchmark is described in Section B´.2.

Additionally, a scoring algorithm is provided to evaluate the results of the different algorithms using various metrics based on the confusion matrix, such as accuracy, precision, recall, F1-score, as well as the detection delay. The evaluation metrics used are explained in Section B´.3. To demonstrate the application of these metrics, a baseline detection algorithm developed by the authors is examined, which is based the concept of Minimum Night Flows (MNF) [71]. The evaluation of this algorithm is provided in Section B´.4. Finally, discussion and future work on this benchmark is provided in Section B´.5. The dataset is stored in an open research data repository given in Section B´.6. The source code of the toolkit that generates the leakage benchmark dataset, as well as the detection algorithm based on MNF are released under the EU Public License.

B´.2 Description of Benchmark

The Leakage Diagnosis Benchmark (LeakDB) was created using the Water Network Tool for Resilience (WNTR), which is an open source Python package designed to help water utilities investigate resilience of water distribution systems to hazards [131]. The variations in different scenarios of the dataset are described below.

The networks in the dataset were carefully selected to have multiple varying characteristics in terms of size, topology and types of elements they contain. Varying the network size, i.e. the number of nodes and links, demonstrates the scalability of the algorithm. Networks with a different number of tanks, reveal the ability of detection algorithms to deal with dynamic states. Topology is also considered, specifically the complexity that arises in leakage diagnosis when the networks contain loops. This is quantified by calculating the circuit rank of the network, which is then normalized by the circuit rank of the same network if it was fully connected. The resulting metric is defined as the *Loop Ratio*, which has a value of zero when there are no loops in the network, while it is equal to one in the case of a fully connected network graph. A table with the benchmark networks used and their characteristics is provided within the LeakDB.

[1] http://archive.ics.uci.edu/ml/

[2] https://github.com/EpistasisLab/penn-ml-benchmarks

The models of the networks used in the dataset are provided in the form of an EPANET 'INP' file. In different scenarios of the LeakDB these networks vary in terms of model parameters, thus introducing modeling uncertainty. These parameters are the pipe length, diameter and roughness, pump curves and settings and tank dimensions, which were varied in a range between $\pm\mu\%$ of their original value. In this work, this range was set to $\mu = 25\%$. The uncertainties are considered bounded and these bounds are provided in the dataset in the form of a percentage with respect to the provided model. Topological uncertainty is introduced in the form of unknown valve settings, i.e. at each scenario the initial status of n pipes (closed or opened) is randomly varied in respect to the provided model. The modified network model for each scenario is also provided in the dataset.

The loading conditions of the network can greatly affect the performance of algorithms, so different demand scenarios at the nodes of the network are considered. The demands are artificially created based on historical real-data from water utilities, which were decomposed into three signal components: (a) weekly periodic component, (b) yearly seasonal changes component and (c) random component [71]. The weekly periodic component describes the fluctuation of demand signals throughout one week, and depends on various social and economic characteristics of the consumers. The yearly seasonal changes component describes the variation in water consumption as a result of seasonality within a year. The random component considers the high frequency variations, which may be due to unpredictable consumer demands, transients, repairs and other network activities. All components are normalized around a zero mean in the range of $[-1, 1]$. Long-term trends are not considered in demand generation, as each scenario spans one year.

The creation of unique demand signals is based on the extracted components from real-data and a multiplicative formulation with respect to the signal components [196]. The two basic periodic components are approximated using Fourier Series (FS), as shown in Figure B′.1. For the weekly periodic component (a), a 20^{th} order FS approximation is used with period of one week. For the yearly seasonal changes component (b), a 3^{rd} order FS approximation is used with a period of one year. To create unique demand patterns that also resemble the original real patterns, the calculated FS coefficients are randomized in a range of $\pm\delta\%$ of their original value, where in this work $\delta = 10\%$. The resulting weekly periodic component and yearly seasonal component for each node are indicated by C_w and C_y respectively. The unique random component C_r is created by generating variables with a normal distribution around a zero mean and a standard deviation equal to ϵ, where in this work $\epsilon = 0.33$. Additionally, a base demand β is calculated by randomizing around an appropriate base demand value specific to each network. Finally, the unique demand pattern d for each node is generated by multiplying all components as follows: $d = \beta(C_w + 1)(C_y + 1)(C_r + 1)$. Note that the simulated demands in each scenario may be different from the demand pattern, due to the use of pressure-driven solver. All scenarios ensure that the network operates normally, given minimum pressure requirements.

Leakages are considered the only type of fault that can occur in the scenarios of the LeakDB dataset. These are simulated at network nodes and at each scenario the location, start time, end time and number of leakages is selected at random. The leakage magnitude varies due to the assignment of a random leakage hole diameter ϕ, which in this work was varied such that $\phi \in [2\text{cm} - 20\text{cm}]$, as well as due to the pressure at the leakage location. Each leakage is also assigned a time profile, categorizing them as *abrupt* leakages, or *incipient* leakages. Incipient leakages increase gradually and are harder to detect by many algorithms that use past behavior to define anomalies in data, as they consider a new behavior can be anomalous at first but ceases to be anomalous if it persists; i.e. a new normal pattern is established. All information regarding the leakages occurring at each scenario is included in the dataset.

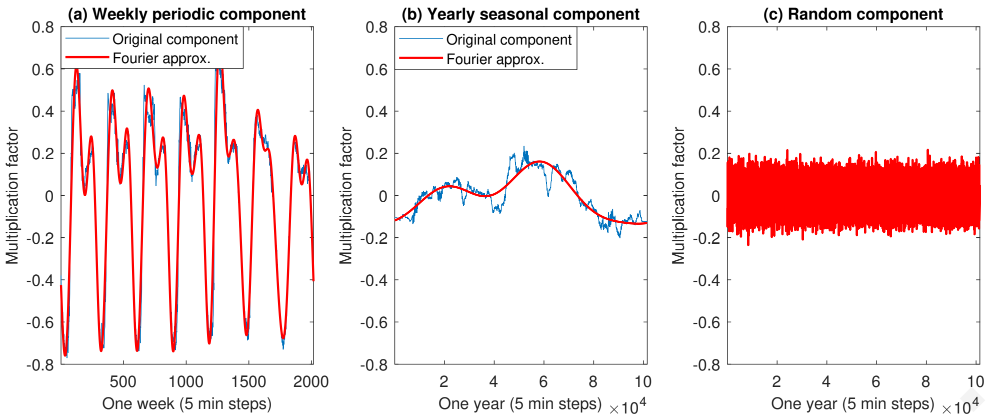


Figure B´.1. Demand components for demand time series construction

The LeakDB dataset divides the simulated scenarios of each network into different directories, so that network specific algorithm evaluations can be performed. Each *Network Directory* contains n_{sc} number of scenarios (e.g., 500 scenarios), with different number of leakage events (or no events). In each *Scenario Directory*, the modified network used is provided in EPANET 'INP' format. Additionally, for each scenario the node demands, node pressures, link flows and leak flow and info are provided in 'CSV' data files. Each data file has the same structure, containing the "Timestamp" column (`YYYY-MM-DD hh:mm:ss`) and a "Value" column (real numbers). The structure of the LeakDB and data file format are illustrated in Figure B´.2.

The LeakDB dataset is accompanied by *Label* files, indicating faults in binary notation. A Label file in each Network Directory, tags individual scenarios indicating if they contain a fault. Within each Scenario Directory, a Label file tags each individual time step of the scenario in which a fault exists. In the folder containing leakage information of each scenario, the start and end time of each individual fault, the leak hole diameter and leak type are provided.

B´.3 Scoring Real-Time Leakage Detectors

The ideal leakage detection algorithm will have the following characteristics [194]: (C1) Detects all leakages present in the streaming data; (C2) Triggers no false alarms (no false positives); (C3) Detects leakages as soon as possible, ideally before the leakage becomes visible to a human; (C4) Works with real time data (no look ahead); (C5) Is fully automated across all datasets (parameter learning should be performed online). While characteristics C4 and C5 can either apply or not apply to a given algorithm, characteristics C1-C3 must be evaluated for each algorithm.

B´.3.1 Classification ability score

In terms of classification accuracy, the scoring algorithm uses the data labels provided with the LeakDB to calculate the standard classification metrics of the confusion matrix: the true positive (TP), false positive (FP), true negative (TN), and false negative (FN). These are calculated for each time step of each scenario as follows:

$$TP = \sum_{i=1}^{n_{sc}} \sum_{k} TP_i(k), \tag{B´.1}$$

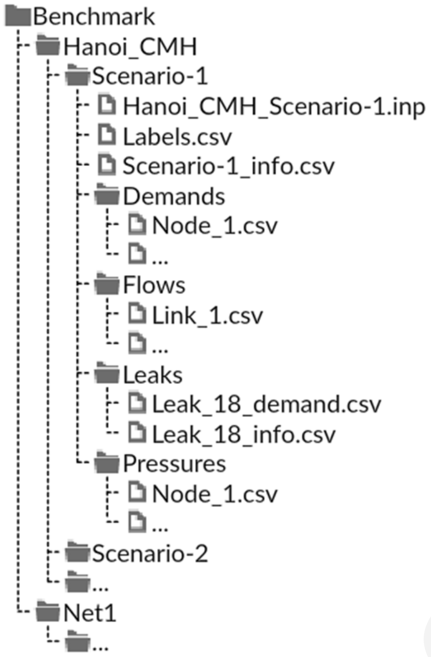


Figure B΄.2. Structure of the LeakDB dataset and an example data file

with $TP_i(k)$ being a true positive in the i-th scenario on the k-th time step and n_{sc} is the number of scenarios considered. The rest of the elements of the confusion matrix TN, FP, FN are calculated in the same manner. Widely used metrics such as precision, recall, specificity and accuracy, can be computed based on the confusion matrix. Using these metrics, the ability of the algorithm to detect leakages in conditions where a leakage exists (C1) is quantified using the TP Rate (or recall) for all time steps in all scenarios, defined as:

$$\mathrm{R_{TP}} = TP/\left(TP + FN\right). \tag{B΄.2}$$

Additionally, the ability of the algorithm to avoid false alarms during the absence of leakages (C2) is quantified using the TN Rate (or specificity) for all time steps in all scenarios, defined as:

$$\mathrm{R_{TN}} = TN/\left(FP + TN\right). \tag{B΄.3}$$

In addition to the above, the F-measure (F1-score) is considered, which balances precision and recall of classifiers on each class by calculating their harmonic mean. The F-measure is a popular metric in datasets which are characterized by class imbalances. Class imbalance occurs when the total number of a class of data (positive) is significantly less than the total number of another class of data (negative). This problem is common in practice and can be observed in various disciplines including anomaly detection. The metric can be calculated separately by considering the confusion matrix computed for each scenario, or, by computing the total number of TP, FP and FN across all scenarios; the latter was shown to be unbiased in the case of class imbalances [197]. This is calculated as :

$$F_{tp,fp} = 2TP/\left(2TP + FP + FN\right). \tag{B΄.4}$$

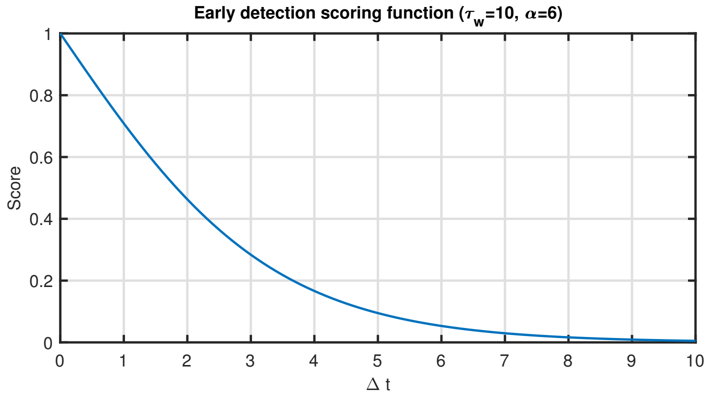


Figure B´.3. Early detection scoring function for a fault window $\tau_w = 10$ time steps.

B´.3.2 Early detection score

The classification scores cannot evaluate effectively real-time detection algorithms as they do not incorporate time and do not reward early detection. Early detection (C3) in this work is evaluated using the *early detection score* which rewards detections that are close to the start time of a fault t_f, while the reward declines as the detection moves further away [194]. The detection time t_d is defined as the earliest time step at which the algorithm registered a detection, inside the fault window duration τ_w. In this work a successful detection is registered if the algorithm gives detections that persist for at least P% of the fault window. The threshold percentage P% is a tuning parameter that prevents algorithms that generate random detections from getting a high early detection score. In this work, the threshold percentage was set to $P\% = 75\%$. The early detection score also considers detections that occur soon after the end of a fault. In order to facilitate this functionality, the fault window duration τ_w is extended for a short time τ_{FP} after the fault ends, where a small positive score is given for a FP that occurs during this time. The extension time is a design parameter which in this work was set to $\tau_{FP} = 5$ hours after the end of a leakage.

The early detection score is calculated by first finding the delay of the first detection in the defined fault window, given by $x = t_d - t_f$. Then the following sigmoid function, also illustrated in Figure B´.3, is used to assign a score to this detection: $\sigma(x) = 2/\left(1 + e^{((\alpha/\tau_w)x)}\right)$, where $x \in \{0, \cdots, \tau_w\}$. The parameter $\alpha > 0$ is defined such that $\sigma(\tau_w) \approx 0$, while $\sigma(0) = 1$ for any value of $\alpha < \infty$. Here a value of $\alpha = 6$ was chosen. A detection that occurs outside the fault window τ_w is not included in the score. The *raw* early detection score is then given by the sum of the early detection function score for all the faults f_j, $j \in \{1, \cdots, n_f\}$ that occur in each scenario $i \in \{1, \cdots, n_{sc}\}$, as follows:

$$S^*_{ED} = \sum_{i=1}^{n_{sc}} \sum_{j=1}^{n_f} \sigma(x_{ij}). \tag{B´.5}$$

The S^*_{ED} then needs to be normalized by the score of the ideal algorithm on this metric, given by S^{ideal}_{ED}. In contrast, the rest of the metrics have ideal scores equal to one, thus the calculated scores are already normalized. The *normalized* early detection score is then given by:

$$S_{ED} = S^*_{ED}/\left(S^{ideal}_{ED}\right) \tag{B´.6}$$

B´.4 Evaluation of detection algorithms

To demonstrate the usage of the LeakDB dataset, four detectors were used which can be considered as a baseline for testing other detectors: An “Ideal” detector which assumes

Detectors	Score (%)			
	R_{TP}	R_{TN}	$F_{tp,fp}$	S_{ED}
Ideal	100.00	100.00	100.00	100.00
Null	0.00	100.00	0.00	0.00
Rand	50.19	49.79	32.11	0.21
MNF 10%	68.98	66.96	50.01	45.88
MNF 20%	39.90	99.58	56.49	32.35
MNF 30%	32.87	99.91	49.36	26.80

Table B´.1. Example detectors and their scores when applied on the LeakBD dataset of the "Hanoi" network. With respect to the F-measure, the best classifier is the MNF with $\lambda = 20\%$.

perfect knowledge and returns correct detections, a "Null" detector which does not detect any faults at any scenario, a "Random" detector which detects faults at random time steps, and a Minimum Night Flow (MNF) detector developed by the authors, using three different detection parameters. To test these detectors the benchmark dataset for 'Hanoi' network was used, by simulating $n_{sc} = 500$ leakage scenarios. The duration of each scenario is one year, with a time step of 30 minutes.

The MNF detector analyzes the minimum night flows during the night, to detect anomalies on the input flow of the network. A moving window w is defined in order to calculate the minimum MNF (mMNF) during those days; this is defined as $\text{mMNF}(k, w) \triangleq \min\{\text{MNF}(\tau)\}$ subject to $\tau \in \{k-w-1, \cdots, k-1\}$. Then, the MNF of the k-th day $\text{MNF}(k)$ is computed and the difference $e(k) = \text{MNF}(k) - \text{mMNF}(k, w)$ is calculated. A fault is detected if the difference $e(k)$ is greater than a threshold as follows: $e(k) > \lambda\ \text{mMNF}(k, w)$. Both the w parameter and $\lambda \in (0, 1)$ are design parameters which can be chosen with the help of the LeakDB dataset. When the threshold is exceeded, an error flag is raised, and the measurement is excluded from the calculation of the mMNF, i.e., only the last w measurements that were not flagged as faults are considered. This helps in avoiding learning the fault after the period w has passed. The MNF detector was scored when applied to the LeakDB scenarios using three different threshold gains: $\lambda = \{10\%, 20\%, 30\%\}$, and a moving window w corresponding to 10 days. The results for all the detectors are shown in Table B´.1.

The results in Table B´.1 indicate that the MNF algorithm performs better than a random algorithm. The MNF detector performs poorly on the early detection metric, as it can only detect an anomaly in the data during the night hours when minimum flow conditions occur. The comparison of the MNF algorithm using different thresholds, shows that there is a tradeoff between early detection and classification accuracy. In this work we do not provide an overall score to the detection algorithms, so the "best" threshold for this algorithm can only be defined with respect to a specific metric. However, to choose the "best" leakage detection algorithm, appropriate weights must be given to each of the provided metrics, or use multi objective optimization methodologies.

B´.5 Conclusions and future work

This benchmark is currently being expanded, as more networks, detection algorithms and features are added. Future work on the benchmark can include additional features such as: 1) considering various anomalies that are due to sensor measurements such as spikes in data, missing measurements, sensor faults and noisy measurements, 2) consider other system faults, such as failure of a pump, 3) include real-world networks and data, in which case an important task would be for experts to manually inspect the data and label the faults.

During the creation of the benchmark, many challenges arose that should be addressed in future work. Firstly the benchmark space requirements may be excessive especially for large networks. This can be addressed using different data formats or omitting redundant data. Additionally, the various scenarios created need to be automatically validated in terms of satisfying realistic hydraulic conditions. Finally, the selection of an overall evaluation metric for the detection algorithms must be investigated.

B΄.6 Data availability

The source code of the toolkit to generate the Leakage Diagnosis Benchmark, as well as links to the datasets, can be found at: https://github.com/KIOS-Research/LeakDB.