Advancing Urban Flood Resilience with Smart Water Infrastructure

by

Matthew Bartos

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Branko Kerkez (Chair)
Professor Alfred Hero
Associate Professor Valeriy Ivanov
Associate Professor Jeffrey Scruggs

One must have a mind of winter
To regard the frost and the boughs
Of the pine-trees crusted with snow;

And have been cold a long time
To behold the junipers shagged with ice,
The spruces rough in the distant glitter

Of the January sun; and not to think
Of any misery in the sound of the wind,
In the sound of a few leaves,

Which is the sound of the land
Full of the same wind
That is blowing in the same bare place

For the listener, who listens in the snow,
And, nothing himself, beholds
Nothing that is not there and the nothing that is.

— WALLACE STEVENS, The Snow Man

Matthew D. Bartos

mdbartos@umich.edu

ORCID: 0000-0001-6421-222X

Dedicated to B.J.B.

Acknowledgments

I would like to thank my advisor Dr. Branko Kerkez for his guidance over the course of my graduate studies. In addition to being an innovative researcher and a skilled engineer, he is also a genuinely kind person and I am truly lucky to have worked with him. I would also like to thank the members of my dissertation committee: Dr. Alfred Hero for encouraging my research into complex networks; Dr. Jeffrey Scruggs for helping to inculcate an appreciation for theory; and Dr. Valeriy Ivanov for grounding me within the tradition of the hydrologic sciences.

I would also like to thank my co-authors, including Brandon Wong for his invaluable work on the *open storm* project; and Hyongju Park, Tian Zhou, and Ram Vasudevan for their major contributions to our paper on vehicle-based rainfall sensing. Without them, this dissertation would not have been possible.

Finally, I would like to acknowledge my funding sources, including the National Science Foundation (grants 1442735, 1639640 and 1737432), the University of Michigan (MCubed grant 985), the Earth Science Information Partners, and the J. Robert Beyster Computational Innovation Graduate Fellows Program.

Table of Contents

Chapter

List of Figures

List of Tables

Abstract

Advances in wireless communications and low-power electronics are enabling a new generation of smart water systems that will employ real-time sensing and control to solve our most pressing water challenges. In a future characterized by these systems, networks of sensors will detect and communicate flood events at the neighborhood scale to improve disaster response. Meanwhile, wirelessly-controlled valves and pumps will coordinate reservoir releases to halt combined sewer overflows and restore water quality in urban streams. While these technologies promise to transform the field of water resources engineering, considerable knowledge gaps remain with regards to how smart water systems should be designed and operated. This dissertation presents foundational work towards building the smart water systems of the future, with a particular focus on applications to urban flooding. First, I introduce a first-of-its-kind embedded platform for real-time sensing and control of stormwater systems that will enable emergency managers to detect and respond to urban flood events in real-time. Next, I introduce new methods for hydrologic data assimilation that will enable real-time geolocation of floods and water quality hazards. Finally, I present theoretical contributions to the problem of controller placement in hydraulic networks that will help guide the design of future decentralized flood control systems. Taken together, these contributions pave the way for adaptive stormwater infrastructure that will mitigate the impacts of urban flooding through real-time response.

Chapter 1

Introduction

Urban flooding is one of the most critical unsolved challenges in contemporary civil engineering. Floods are the leading cause of natural disaster fatalities worldwide, with roughly 540,000 deaths attributable to floods between 1980 and 2009 [1]. In addition to their direct toll on human lives, floods inflict severe economic costs by damaging homes [1], collapsing bridges [2], and releasing harmful pollutants into waterways [3–5]. Urban floods are particularly hazardous due to their proximity to population centers and critical infrastructure [6]. Already the most costly natural disaster, urban floods are expected to worsen in coming decades as cities contend with aging infrastructure [7], increasing urbanization [8], and more frequent extreme weather events [9]. Despite the serious societal impacts posed by urban flooding, the physical dynamics that govern these events are still poorly understood [6]. Modeling of urban flooding is complicated by highly heterogeneous flow paths and the presence of stormwater infrastructure [6, 10]. Flash floods caused by excessive localized rainfall are especially difficult to model or forecast, given that the flood response is extremely rapid and sensitive to spatial variabilities in rainfall [6]. To date there is no existing model that is "capable of making reliable flash flood forecasts in urban watersheds" [6], making the development of effective response measures difficult.

Traditionally, engineers have addressed urban flooding by expanding *gray* infrastructure such as retention basins and conveyance pipes [11]. Larger retention basins help to store excess stormwater during wet weather events, while larger conveyance pipes help to carry stormwater away from sensitive urban areas. However, new construction comes with a number of disadvantages. First, building new stormwater infrastructure is expensive, and many cash-strapped cities lack the funds needed for periodic upgrades [12]. Second, expanding stormwater infrastructure often results in adverse environmental side effects [13]. Oversized stormwater pipes may increase discharges into natural streams and lakes, causing erosion and damaging aquatic ecosystems [13–17]. Finally, expansion of passive infrastructure often fails to keep pace with changing hydrologic, demographic, and regulatory demands [18]. Cities may spend considerable sums of money on a new stormwater system, only to find that it no longer meets their needs.

Smart stormwater systems offer a new approach to address urban flooding while solving many of the limitations of static infrastructure [19]. Drawing on recent advances in low-power sensing and embedded controls, these systems transform passive stormwater networks into active ones—allowing operators to preemptively detect water hazards and then reconfigure infrastructure to mitigate impacts in real-time. Networks of high-resolution depth sensors will one day allow emergency managers to detect flood events at the level of individual roadways, enabling targeted alerts for motorists, improved dispatch of emergency services, and adaptive routing of traffic around inundated areas. Taken even further, these systems will enable operators to actively control stormwater infrastructure to mitigate flooding impacts. By coordinating water releases across cities using wirelessly-controlled valves and pumps, smart stormwater systems will halt floods and combined sewer overflows before they occur. While the technologies are new, pilot studies show that these systems are capable of achieving significant performance gains compared traditional steel-and-concrete infrastructure at a fraction of the price [12, 20–26].

While smart water systems offer a revolutionary vision for the future of water resources management, new research is needed to translate this vision into reality. Real-world adoption of smart water systems is currently limited by a number of considerable knowledge gaps:

- First and foremost, the fundamental hardware and software infrastructure needed to instrument stormwater systems for real-time sensing and control does not currently exist. *How can we develop a new embedded platform to enable scalable real-time monitoring and control of urban drainage networks?*

- Real-time response to urban flooding requires extremely high resolution estimates of rainfall that are not available in most cities. *How can we leverage new data sources to better characterize rainfall over urban areas?*

- Sensor data alone is not sufficient to understand or control the dynamics of urban drainage networks. *How can we reconcile real-time sensor data with our best scientific understanding of stormwater systems, as represented by physically-based models?*

- The impact of flooding on urban water quality is poorly understood due to a lack of real-time sensor data and appropriate process models. *How can we better understand the fate and transport of contaminants during urban flood events?*

- With respect to dynamic control of stormwater systems, there are few established guidelines on how control sites should be selected. *How can we select control sites in order to maximize their effectiveness?*

My dissertation will address these knowledge gaps by introducing new scientific advances in distributed sensing, data assimilation, and control of stormwater infrastructure. In addition to improving our fundamental understanding of flooding in urban drainage systems, this work will provide practical tools that will enable water managers to better monitor and respond to urban flood events in real-time. This work is divided into the following five chapters.

Chapter 2: This chapter details the development of an end-to-end hardware/software platform that combines embedded systems, wireless communications and cloud services to enable continuous monitoring and control of stormwater systems at unprecedented scales.

Chapter 3: This chapter presents a new methodology for improving rainfall field estimates using windshield wiper data from connected vehicles.

Chapter 4: This chapter introduces a novel methodology for real-time state estimation in sewer/channel networks using a custom-built hydraulic solver alongside an implicit Kalman filtering approach.

Chapter 5: This chapter describes a new numerical scheme for real-time modeling of contaminant fate and transport in urban drainage networks using the advection-reaction-diffusion equation.

Chapter 6: This chapter advances a new graph-theoretic algorithm for placing hydraulic control structures within drainage networks that maximally attenuates peak flows.

Chapter 2: An end-to-end platform for real-time monitoring and control of urban water systems

Advances in *internet of things* technologies are transforming water resources management by enabling dynamic monitoring and control of previously static infrastructure [19]. However, adoption of *smart* water systems has been hampered by a lack of "plug-and-play" solutions that combine the necessary hardware, software and documentation into an accessible package. In this chapter, a new end-to-end platform for real-time monitoring and control of urban water systems is developed that will enable cities to build and maintain their own *smart* stormwater systems.

In spite of the many benefits offered by smart stormwater systems, mainstream adoption has been slow [19]. A major factor in this delay is the lack of purpose-built systems for instrumentation and control. While some proprietary solutions exist [12], there are no open-source hardware/software platforms designed specifically for real-time monitoring and control of urban drainage networks. Moreover, acceptance of *smart* infrastructure within the water resources community has been hindered by a lack of real-world case studies that document the steps needed to deploy these systems and demonstrate the tangible benefits that can be achieved.

To address these knowledge gaps, this chapter introduces *open storm*—a new hardware and software platform for real-time sensing and control of urban watersheds. First, a robust, low-power wireless sensor node is developed that interfaces with common sensors and actuators for water applications. The sensor node is then integrated with cloud services to enable system-level supervision and control of hydraulic infrastructure. The effectiveness of the system is then demonstrated through two real-world case studies: a flood-monitoring network in the Dallas–Fort Worth metroplex, and a real-time stormwater control system in Ann Arbor.

This chapter makes several fundamental contributions to the field of environmental sensing. The *open storm* framework described in this chapter is the first open-source, end-to-end hardware and software platform that is specifically designed for distributed monitoring and control of urban water systems. This platform has been adopted by researchers at several universities, including Yale University, the University of Wisconsin–Madison, and the University of Pittsburgh. The sensor networks deployed in this chapter will serve as the basis for long-term experiments in real-time data assimilation and control that will be explored in later chapters of this dissertation.

Chapter 3: Enhancing rainfall predictions using windshield wiper data

Accurate precipitation measurements are essential for virtually every aspect of water resources management [27]. However, conventional methods for measuring rainfall are subject to spatial and temporal uncertainties that render them unsuitable for high-precision applications like flash flood forecasting [6]. Vehicle-based measurements offer a novel way to fill gaps in these data sources by providing precise information about the timing and location of rainfall. This chapter paves the way for an entirely new field of vehicle-based environmental sensing by showing that measurements from connected vehicles can significantly enhance the quality and coverage of rainfall estimates.

While accurate rainfall data are required at high resolutions for applications like flash-flood forecasting, weather radar and stationary gages are both highly uncertain [6]. Stationary gage networks are typically too sparse to resolve spatial patterns in rainfall [28–31], while weather radar suffers from thresholding issues and uncertain reflectivity-rainfall relationships [32–35]. One method for improving the resolution of rainfall estimates is to fuse these traditional data sources with measurements from *ubiquitous sensors* located on consumer products [36]. Connected vehicles offer an as-of-yet untapped source of environmental data via incidental onboard sensors such as windshield wipers. However, to date no existing studies have examined the use of windshield wipers as rainfall sensors using real data [37, 38].

To address this knowledge gap, this study is the first to use real-world vehicle data to evaluate the effectiveness of windshield wipers as rainfall sensors. Drawing on data from roughly 70 connected vehicles, this study evaluates the effectiveness of windshield wipers as rainfall sensors by comparing measured wiper intensities to dashboard camera footage representing the ground truth rainfall state. A new Bayesian filtering methodology is then derived to generate updated rainfall maps that incorporate both weather radar and binary wiper observations. The binary detection accuracy of this new data product is evaluated using a leave-one-out cross-validation approach.

This chapter makes two major contributions. First, this study is the first to examine the use of windshield wipers as rainfall sensors using real-world data. Through analysis of a large real-world connected vehicle dataset, this study finds that windshield wiper measurements predict the binary rainfall state (raining/not raining) with greater accuracy than either weather radar or rain gages. Second, by combining wiper observations and weather radar using a Bayesian filter, this study creates a new data product that captures the spatial variability of rainfall more effectively than weather radar alone.

Chapter 4: Real-time data assimilation in urban stormwater networks using an implicit hydraulic solver with Kalman filtering

New sensing and communication technologies are providing utilities with increased access to real-time data. However, integration of real-time data into stormwater system operations is limited by a lack of appropriate models and data assimilation techniques. New methods are needed to improve estimation, interpolation, and forecasting of system states while at the same time respecting the physical dynamics of stormwater networks. In this chapter, I present a new methodology for performing online data assimilation in urban stormwater networks by combining a custom-built hydraulic solver with an implicit Kalman filtering approach.

Within the stormwater industry, many utilities are seeking to build *digital twins* that combine sensor data with online models in order to better understand system dynamics, detect maintenance emergencies, and implement real-time control [39]. However, real-world implementation of these systems has been hampered by a lack of suitable hydraulic models and a lack of theory regarding how data should be assimilated. Most stormwater models are designed to be executed in batch rather than online mode, and do not allow for internal states to be modified during execution. Second, existing stormwater models typically do not admit a state-space representation of system dynamics, making it difficult to apply classical state estimation techniques such as Kalman filtering. As a result, few studies have examined the application of formal data assimilation techniques to urban drainage systems, and efforts to build *digital twins* of stormwater networks have largely centered on continuous calibration of offline models.

To address these knowledge gaps, this chapter introduces the first *digital twin* model for urban drainage networks that is capable of fusing real-time sensor data into a process model based on the full one-dimensional Saint Venant equations. First, I formulate and implement a new implicit hydraulic solver that is capable of online operation and also yields a state-space model of the sewer/channel system [40]. Next, I derive a method for assimilating real-time data into the dynamical model using an implicit Kalman filtering approach. Finally, drawing on real-world sensor data collected as part of Chapter 2, I evaluate the framework by showing that data assimilation improves the interpolation and forecasting of system states.

This chapter makes several major contributions to the field of hydrologic data assimilation and real-time operation of stormwater systems. Most notably, this study is the first to apply Kalman Filtering to fuse real-world sensor data into an urban drainage model based on the full one-dimensional Saint Venant equations. The methodology pioneered in this chapter will enable system operators to better detect floods at ungaged locations, identify maintenance emergencies, and more confidently implement real-time control. In addition to its more fundamental scientific contributions, this study also provides a powerful set of tools to the stormwater community by incorporating the hydraulic solver and data assimilation framework into a fully-featured open-source software library aimed at practitioners.

Chapter 5: A new water quality model for urban drainage networks

Urban flooding poses major challenges for urban water quality. However, traditional methods for evaluating contaminant fate and transport are often insufficient for the heterogeneous structure and complicated flow regimes of stormwater networks. This chapter introduces a new numerical scheme for modeling contaminant fate and transport within urban drainage networks using the one-dimensional advection-reaction-diffusion equation.

Contaminant fate and transport within stormwater networks can be described by the one-dimensional advection-reaction-diffusion equation [41]. In the general case, this equation can prove difficult to solve due to numerical instability and loss of accuracy at coarse discretizations [42]. However, for the particular case of urban stormwater networks, the solution of this equation is made even more difficult by the inherent heterogeneity of urban drainage systems (which may consist of pressurized pipes, open channels, storage elements, and various types of control structures). For these reasons, stormwater modelers typically rely on heuristic methods for simulating water quality [43]. However, these methods may provide misleading results, or prevent results from being properly interpreted.

To address these knowledge gaps, this chapter introduces a new numerical scheme specifically formulated to produce stable solutions to the advection-reaction-diffusion equation in urban drainage networks. This numerical scheme uses a staggered-grid implicit formulation to ensure stability over large time steps. The numerical method is tested against analytical solutions to the advection-reaction-diffusion equation, and is found to maintain accuracy and stability at large time steps.

The numerical scheme introduced in this chapter will provide stormwater engineers with new tools for better understanding contaminant fate and transport in urban drainage networks. In particular, the stability afforded by this new numerical scheme will enable system operators to run continuous online simulations of urban water quality, allowing them to better understand how stormwater systems are performing in real-time. Moreover, because the numerical scheme admits a state-space representation, it will enable system operators to apply techniques from control theory to better understand and control water quality—for instance, applying Kalman Filtering to fuse sensor data and detect contaminant spread in real time.

Chapter 6: A graph-theoretic algorithm for placing hydraulic control structures in drainage networks

Hydrograph peak attenuation is a central goal in the design of urban stormwater networks. By building systems of distributed reservoirs, stormwater engineers can stabilize the hydrologic response of urban watersheds, thereby mitigating flash floods, limiting streambed erosion, and reducing contaminant loads [19]. While the location of control sites affects the performance of these systems, few guidelines exist for siting hydraulic control structures in order to best mitigate impacts. To address this problem, this chapter presents a new algorithm for determining the optimal placement of hydraulic control structures in drainage networks.

As water managers increasingly move towards decentralized stormwater solutions, efficient placement of hydraulic control structures will become more important for ensuring maximum flood control and water quality benefits [44]. However, within the literature there is currently little guidance on how stormwater control sites should be selected. Previous studies exploring controller placement in drainage networks focus on individual case studies, and generally approach the problem through exhaustive optimization [45–48]. Little research has been done to link control of distributed reservoir systems with structural properties of the underlying drainage network [49]. In short, there is a need for a new theoretically-based approach for placing hydraulic control structures in drainage networks.

To address these knowledge gaps, this chapter presents a new controller placement algorithm that reduces peak discharges by controlling the locations in the drainage network that maximally de-synchronize tributary flows. Specifically, the algorithm uses the catchment width function to find the "cuts" in the drainage network that remove regions of the watershed that contribute disproportionately to the peak of the hydrograph. Through an ensemble of hydrodynamic simulations, it is shown that the algorithm attenuates hydrograph peaks more effectively than randomized controller placement strategies.

In terms of its major contributions, this study is the first to provide a theoretically-motivated solution to the problem of optimal controller placement in drainage networks. This algorithm will provide stormwater engineers with the tools to design future stormwater control systems that achieve more robust flood control and water quality benefits. Moreover, the algorithm presented in this chapter is fast and requires only digital elevation data, making it accessible for practitioners.

Chapter 2

Open Storm: A Complete Framework for Sensing and Control of Urban Watersheds

Abstract

Leveraging recent advances in technologies surrounding the *Internet of Things*, "smart" water systems are poised to transform water resources management by enabling ubiquitous real-time sensing and control. Recent applications have demonstrated the potential to improve flood forecasting, enhance rainwater harvesting, and prevent combined sewer overflows. However, adoption of smart water systems has been hindered by a limited number of proven case studies, along with a lack of guidance on how smart water systems should be built. To this end, we review existing solutions, and introduce *open storm*—an open-source, end-to-end platform for real-time monitoring and control of watersheds. *Open storm* includes (i) a robust hardware stack for distributed sensing and control in harsh environments (ii) a cloud services platform that enables system-level supervision and coordination of water assets, and (iii) a comprehensive, web-based "how-to" guide, available on `open-storm.org`, that empowers newcomers to develop and deploy their own smart water networks. We illustrate the capabilities of the *open storm* platform through two ongoing deployments: (i) a high-resolution flash-flood monitoring network that detects and communicates flood hazards at the level of individual roadways and (ii) a real-time stormwater control network that actively modulates discharges from stormwater facilities to improve water quality and reduce stream erosion. Through these case studies, we demonstrate the real-world potential for smart water systems to enable sustainable management of water resources.

2.1 Introduction

Advances in wireless communications and low-power sensing are enabling a new generation of "smart cities," which promise to improve the performance of municipal services and reduce operating costs through real-time analytics and control [50]. While some applications of "smart" infrastructure have received a great deal of attention—such as autonomous vehicles [51, 52], energy grid management [52], and structural health monitoring [52, 53]—integration of these technologies into water systems has lagged behind. However, "smart" water systems offer new inroads for dealing with many of our most pressing urban water challenges, including flash flooding, aquatic ecosystem degradation, and runoff pollution. The goal of this paper is to provide an end-to-end blueprint for the next generation of autonomous water systems, with a particular focus on managing urban stormwater. Towards this goal, we introduce *open storm*, an open source framework that combines sensing, real-time control, wireless communications, web services and domain-specific models. We illustrate the potential of *open storm* through two real-world case studies: 1) a 2,200 km^2 wireless flood forecasting network in Texas, and 2) an 11 km^2 real-time stormwater control network in Michigan. Most importantly, to encourage broader adoption by the water resources community, this paper is accompanied by extensive supplementary materials on `open-storm.org`, including videos, photos, source code, hardware schematics, assembly guides, and deployment instructions. These materials make it possible for newcomers to implement their own "smart" stormwater systems, without extensive experience in programming or embedded systems design.

2.2 Background

Motivation

Effective management of water supply and water excess are some of the largest engineering problems faced by cities today [54], and in the wake of rapid urbanization, aging infrastructure, and a changing climate, these challenges are expected to intensify in the decades to come [8, 9]. Floods are the leading cause of severe weather fatalities worldwide, accounting for roughly 540,000 deaths between 1980 and 2009 [1]. Fur-

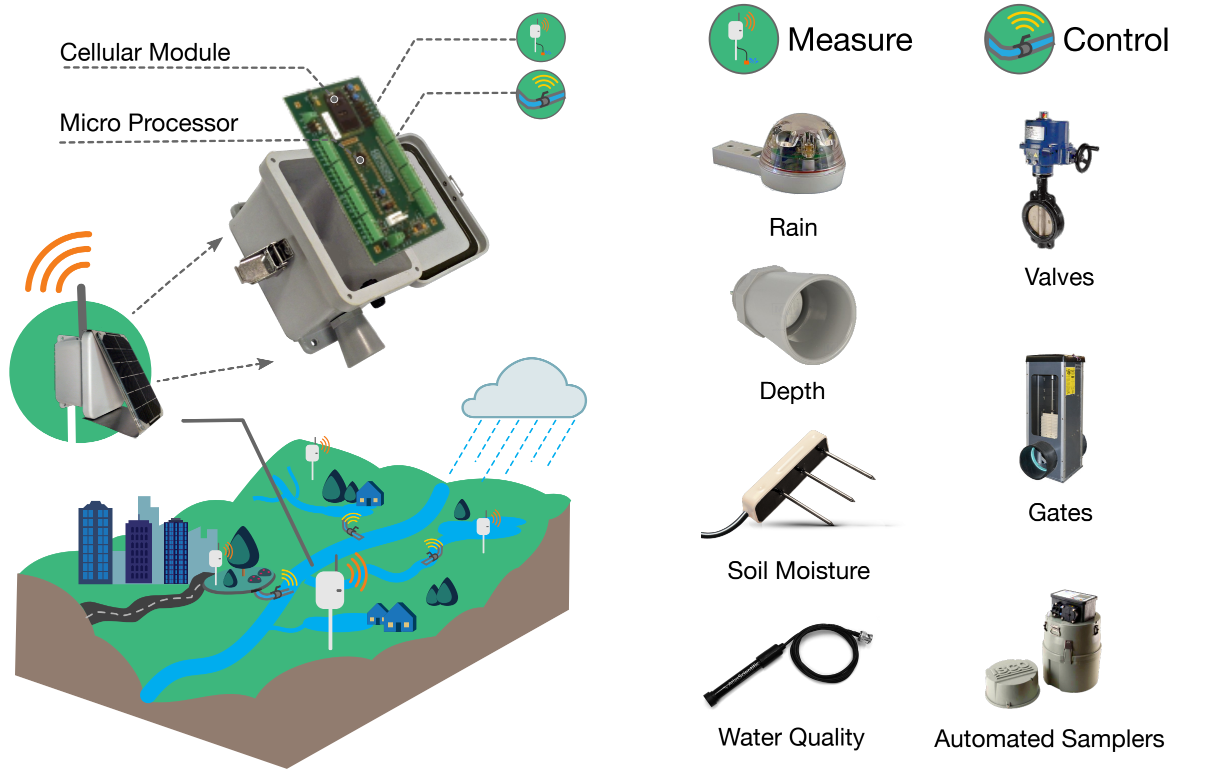


Figure 2.1. The *open storm* hardware layer. The left panel shows the complete sensor node along with a representative schematic of its placement in an urban watershed. The right panel shows typical sensors and actuators used in *open storm* research projects.

thermore, large quantities of metals, nutrients, and other pollutants are released during storm events, making their way via streams and rivers into lakes and coastal zones [3, 4]. The need to manage pollutant loads in stormwater has persistently been identified as one of our greatest environmental challenges [55]. To contend with these concerns, most communities maintain dedicated gray infrastructure (pipes, ponds, basins, wetlands, etc.) to convey and treat water during storm events. However, many of these systems are approaching the end of their design life [7]. At the same time, stormwater systems are being placed under greater stress due to larger urban populations, changes in land use, and the increasing frequency of extreme weather events [9, 54]. In some communities, stormwater and wastewater systems are combined, meaning that they share the same pipes. For these systems, large storms often lead to combined sewer overflows, which release viruses, bacteria, nutrients, pharmaceuticals, and other pollutants into estuaries downstream [5]. When coupled with population stressors, it comes as little surprise that the current state of stormwater infrastructure in the United States has been given a near-failing grade by the American Society of Civil Engineers [56].

Engineers have traditionally responded to increasing demands on stormwater systems by expanding and constructing new *gray* infrastructure. However, the upsizing of pipes and storage elements can prove expensive, time-consuming, and may even result in deleterious long-term side effects. Benefits from stormwater conveyance facilities can be diminished if individual sites are not designed in a global context. Even when best management practices are followed, discharges from individual sites may combine to induce downstream flows that are more intense than those produced under unregulated conditions [57]. Without system-level coordination, gray infrastructure expansion may lead to overdesigned solutions that adversely impact flooding, increase stream erosion, and impair water quality [13]. In response to these concerns, *green* infrastructure (GI) has been proposed as an alternative to traditional "steel and concrete" stormwater solutions. These systems use smaller, distributed assets—such as bioswales, green roofs and rain gardens—to condition flows and improve water quality. However, recent research has raised questions about the scalability and maintenance requirements of green infrastructure [58]. Regardless of the choice between "gray" or "green", new construction is limited by cost, and often cannot keep pace with evolving community needs. To preserve watershed and ecological stability, there is an urgent need to incorporate systems thinking into stormwater designs and to engineer solutions that can optimize stormwater system performance—not only for individual sites, but for entire watersheds.

The promise of sensing and control

"Smart" water systems promise to improve the study and management of water resources by extending monitoring and control beyond centralized facilities and into watersheds as a whole. With increased access to inexpensive sensors and wireless communications, the feasibility of deploying and maintaining large sensor networks across urban landscapes is now within reach for many public utilities and research groups. While many of the technologies have existed for some time, it was not until the integration of wireless sensor networks with web services (i.e. the *Internet of Things*) that large networks consisting of hundreds or thousands of heterogeneous devices could be managed reliably [59]. This in turn has enabled watersheds to be studied at spatial and temporal scales that were previously unattainable. By densely instrumenting urban watersheds, researchers can finally begin to understand the complex and spatially variable feedbacks that govern water flow and quality across the built environment. A system-level understanding of urban watershed dynamics will provide decision makers with actionable insights to alert the public, and improve stewardship of water resources.

Beyond new insight gained through sensing, the ability to dynamically regulate water levels across a watershed will reduce flooding, preserve riparian ecosystems, and allow for distributed treatment of stormwater. While these functions were previously achieved only through construction of static gray infrastructure or centralized treatment facilities, the addition of remotely-controlled valves and pumps promises to realize the same benefits while at the same time reducing costs, expanding coverage, and allowing system performance to scale flexibly with changing hydrologic conditions. Adding valves to existing stormwater facilities, for instance, can extend hydraulic retention time, thereby promoting the capture of sediment-bound pollutants [19, 60]. Modulation of flows (hydrograph shaping) may reduce erosion at downstream locations by ensuring that discharges do not exceed critical levels [19]. More fundamentally, distributed control will enable operators to coordinate outflows from stormwater sites (tens to hundreds) across an entire city. Along with reducing flooding, this will allow water managers to utilize the latent treatment capacity of existing ponds and wetlands—effectively allowing a watershed to function as a distributed wastewater treatment plant [60].

Such a vision for "smart" stormwater systems is no longer limited by technology. Rather, adoption of smart water systems has been hindered by (i) a reliance on proprietary technologies, (ii) a lack of proven case studies, and (iii) an absence of end-to-end solutions that are specifically designed and tested for water resources applications. To enable truly holistic management and control, there is an urgent need to combine modern technologies with domain knowledge from water sciences—something which present solutions do not address or make transparent. These solutions are reviewed next, after which the *open storm* framework is introduced as an end-to-end blueprint for "smart" water systems management. This open-source framework combines low-power wireless hardware with modern cloud computing services and domain-specific applications to enable scalable, real-time control of urban water systems.

2.3 Existing technologies

Real-time sensing and control of water infrastructure is not a new idea. Supervisory control and data acquisition (SCADA) systems have long been used to monitor and control critical water infrastructure [61]. In addition to traditional SCADA systems, there has been a recent explosion in the development of wireless sensor networks (WSNs) for water resources management. While these technologies have made great strides in enabling monitoring and control of water systems, a lack of end-to-end solutions has inhibited system-scale management of watersheds. In this section, we review existing technological solutions for water system monitoring and control, and describe how *open storm* advances the state of the art by providing the first open source, end-to-end solution for distributed water systems management.

SCADA systems

Most water utilities use SCADA systems to manage the conveyance, treatment and distribution of water [61]. These systems comprise collections of devices, communication protocols, and software that enable remote monitoring and control of water assets [61]. Most commonly applied in water distribution systems, SCADA systems typically monitor parameters that indicate service quality—such as flows, pressures, and chemical concentrations—and then use this information to control the operation of pumps and valves in real-time [61]. Control may be manual or automatic, and in some cases may integrate optimization algorithms, decision support systems and advanced control logic [61]. While legacy SCADA systems remain popular among water utilities, they suffer from limitations in three major areas: interoperability, scalability and security.

Perhaps the most critical limitation of legacy SCADA systems is the lack of interoperability between systems, reliance on proprietary protocols, and non-extensible software [62]. Traditional SCADA systems are often isolated and incapable of intercommunication [62]. Systems that manage water in one municipality, for instance, may be incapable of communicating with those in another municipality, despite sharing the same service area. Moreover, different SCADA systems within the same jurisdiction may also be isolated, meaning that management of stormwater systems may not in any way inform the operation of wastewater treatment facilities downstream. This lack of communication between water management architectures makes it difficult to coordinate control actions at the watershed scale. Proprietary SCADA systems are also often unable to interface with modern software layers, like Geographic Information Systems (GIS), network analysis software, or hydrologic models [62]. For this reason, SCADA-based control often cannot take advantage of modern domain-specific tools that would enable system-scale optimization of watershed resources.

The capacity of SCADA systems to implement watershed-scale control is also limited by a lack of spatial coverage. Due to their large power footprint and maintenance requirements, traditional SCADA systems are typically limited to centralized water assets with dedicated line power, such as drinking water distribution systems and wastewater treatment facilities [63]. Sensors are usually deployed at a select few locations within the network—like treatment plants, pump stations and boundaries with other systems—and in many cases plant and pump station discharges are not even recorded [61]. For decentralized applications, such as stormwater networks or natural river systems, the cost and power usage of traditional SCADA systems are prohibitive. As such, these distributed resources often go unmonitored and uncontrolled.

Recent studies have also raised concerns about the security of SCADA systems, many of which were designed and installed decades ago [64, 65]. Many legacy SCADA systems rely on specialized protocols without built-in support for authentication, such as MODBUS/TCP, EtherNet/IP and DNP317 [64, 65]. The use of unsecured protocols means that it is possible for unauthorized parties to execute commands remotely on a device in the SCADA network [64]. To cope with this problem, SCADA networks are often isolated from public networks, such as the internet. However, remote attacks are still possible—particularly through the use of unsecured radio channels [65]. Moreover, isolation from public networks limits the use of modern web services such as cloud computing platforms. Reliance on closed networks and proprietary interfaces may also lend a false sense of security to legacy SCADA systems—a concept known as security through obscurity [64]. For these reasons, SCADA systems have gained the reputation of being relatively closed and only manageable by highly-trained operators or specialized local consultants. While SCADA systems remain the most popular platform for managing urban water systems, new tools are needed to improve security, expand coverage, and encourage integration with modern software.

Wireless sensor networks

The past decade has witnessed a large reduction in the cost and power consumption of wireless electronics; leveraging these advances, wireless sensor networks (WSNs) have opened up new frontiers in environmental monitoring, with applications ranging from biodiversity monitoring [66], forest fire detection [67, 68], precision agriculture [69], glacier research [70], and structural health monitoring [53]. Unlike SCADA systems, WSNs are ideal for low-cost, low-power, and low-maintenance applications, making them well-suited for the monitoring of large water systems like rivers and watersheds. WSNs have been applied to great success in applications ranging from flood monitoring to real-time stormwater control; however, current implementations are generally experimental or proprietary, resulting in a lack of discoverability, limited interoperability, and duplication of effort among projects.

Within the water sciences, flood monitoring represents a particularly important application area for WSNs. While several groups have worked to expand the capabilities of existing legacy flood detection networks [71–73], only a small number of groups have designed and deployed their own flood monitoring WSNs. Hughes et al. (2008) describe a 15-node riverine flood monitoring WSN in the United Kingdom, which interfaces with remote models, performs on-site computation, and sends location-specific flood warnings to stakeholders [74, 75]. Other riverine flood monitoring networks include a 3-node river monitoring network in Massachusetts, a 4-node network in Honduras [76], and—perhaps the largest unified flood monitoring network in the US—the Iowa Flood Information System (IFIS), which draws on a network of over 200 cellular-enabled sensor nodes [77]. While most existing flood-monitoring networks focus on large-scale river basins, flash-flooding has received considerably less attention in the WSN community. Marin-Perez et al. (2012) construct a 9-node WSN for flash flood monitoring in a 660 km^2 semiarid watershed in Spain [78], while See et al. (2011) use a Zigbee-based WSN to monitor gully-pot overflows in an urban sewer system [79]. While most deployments are still pilot-scale, these projects demonstrate the potential of WSNs for distributed flood monitoring across a variety of scales and environments.

In addition to monitoring watershed hazards, a limited—but promising—number of projects are illustrating the potential of WSNs for real-time control. Web-enabled sensor nodes have been used to develop adaptive green infrastructure at a select number of pilot sites—for instance, by using weather forecasts to facilitate predictive rainwater harvesting and capture of sediment-bound pollutants [80]. At larger scales, a combined sewer network in South Bend, Indiana uses over 120 flow and depth sensors along with nine valves to actively modulate flows into the city's combined sewer system [12]. This network optimizes the use of existing in-line storage and has achieved a roughly five-fold reduction in combined sewer overflows from 2006-2014 [12]—all without the construction of additional infrastructure. While distributed control of storm and sewer systems shows promise, most existing implementations are proprietary. A lack of transparency makes these solutions inaccessible to decision makers and the water resources community at large.

Although many research groups have realized the potential for real-time watershed monitoring, existing WSN deployments are generally small-scale and experimental in nature. In order for these networks to be accepted as "critical infrastructure" by the water resources community at large, consistent standards for design, deployment and functionality are needed. In designing their own WSNs, researchers tend to look towards previous research projects [76]. However, research papers rarely include the detailed documentation needed to implement an end-to-end sensor platform [76]. As a result, researchers are often forced to design and deploy their own WSNs from scratch. To prevent duplication of effort and ensure best practices, a community-driven *how-to guide* is urgently needed. Moreover, while proprietary control networks have proven their effectiveness in improving the performance of stormwater systems, an open source alternative is needed to encourage transparency, interoperability, and extensibility. Without open software, standards, and documentation, these new technologies risk becoming like the SCADA systems of old: isolated, proprietary, and incapable of intercommunication.

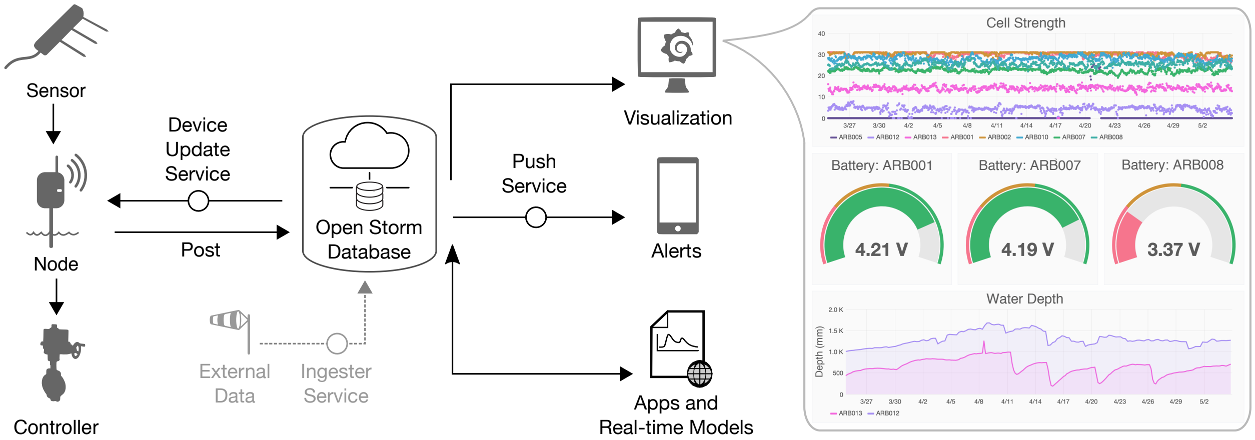


Figure 2.2. The *open storm* stack. The hardware layer (left) comprises the sensor node along with auxiliary sensors and actuators. The cloud services layer (center) includes the database backend, along with a series of publication and subscription services for controlling sensor node behavior and interfacing with applications. The applications layer (right) allows for real-time supervision and control of field-deployed devices. The rightmost panel shows an example dashboard, including sensor feeds and network status visualizations.

2.4 The *open storm* platform

Open storm provides a transparent and unified framework for sensing and control of urban watersheds. To our knowledge, it is the only open-source, end-to-end platform that combines real-time sensing, control and cloud services for the purpose of water resources management. The project is designed to foster engagement by lowering the technological barriers for stakeholders, decision makers, and researchers. To this end, the *open storm* framework is accompanied by a body of reference material that aims to make it easy for non-experts to deploy their own sensors and controllers. This living document, available at `open-storm.org`, provides tutorials, documentation, supported hardware, and case studies for end-to-end sensor network management. In addition to documenting core features, this guide details the (literal) *nuts-and-bolts* of sensor network deployment, including information that is typically not available in journal articles—such as mounting hardware, assembly instructions and deployment techniques.

The *open storm* framework can broadly be divided into three layers: hardware, cloud services, and applications (Figure 2.2). The *hardware* layer includes devices that are deployed in the field—such as sensors for collecting raw data, actuators for controlling water flows, microprocessors, and wireless transmitters. The *cloud services* layer includes processing utilities that receive, store and process data, and interact with field-deployed devices through user-defined applications. Finally, the *application* layer defines how users, algorithms, and real-time models interact with field-deployed devices. This three-tier architecture allows for applications to be developed at a high level, without the need for low-level firmware programming. Together, these layers comprise a scalable framework that can easily be adapted to the needs of a wide variety of users and applications.

Hardware

The sensor node

At its core, the *open storm* hardware layer (Figure 2.1) is enabled by the sensor node—a custom low-power embedded computer with wireless capabilities. The sensor node collects measurements from attached sensors, transmits and receives data from a remote server, and executes control actions. A microcontroller (PSOC5-LP by Cypress Semiconductor) serves at the processing unit for the board. This microcontroller is programmed with a simple operating system that schedules the tasks to be executed, and interfaces with a series of device drivers that control the behavior of attached sensors and actuators. The operating system is designed to minimize power use and consists of a single routine which (i) wakes the device from sleep mode, (ii) downloads pending instructions from the cloud server, (iii) takes sensor readings and triggers actuators, (iv) transmits sensor data to the server, and (v) puts the device back into sleep mode. The sensor node spends the majority of its deployment in sleep mode, allowing it to conserve battery power and remain in the field for an extended period of time.

The sensor node uses wireless telemetry to transmit and receive data from a remote server. While internet connectivity can be achieved through a number of wireless protocols, *open storm* nodes currently use a cellular communications protocol, which enables telemetry through 2G, 3G and 4G LTE cellular networks. Cellular connectivity is implemented through the use of a cellular module (by Telit), along with a small antenna for broadcasting the wireless signal. Compared to other protocols (such as satellite or wi-fi), cellular telemetry is especially suitable for urban and suburban environments due to (i) consistent coverage, (ii) relatively low cost, and (iii) high data throughput. At the time of writing, IoT cellular data plans can be purchased for under $5 per month per node (1-10 MB), making it financially feasible for even small research groups to maintain large-scale networks.

The sensor node is equipped with a power regulation subsystem to provide power to the microcontroller and attached devices. The power supply system consists of four components: (i) a battery, (ii) a solar panel, (iii) a charge controller, and (iv) a voltage converter. The voltage converter permits the sensor node to be powered across a range of 3-40V. While most sensor nodes are powered by a 3.7V Lithium Ion battery, 12V batteries can also be used for higher-voltage sensors and actuators. The solar panel and solar charger are used to recharge the battery, allowing the device to remain in the field without routine maintenance. At the time of writing, many field-deployed sensor nodes have reported data for over a year without loss of power.

Detailed technical information regarding the sensor node—including parts, schematics and programming instructions—are available online at `open-storm.org/node`. Excluding the cost of auxiliary sensors, the sensor node can currently be assembled from off-the-shelf parts for a price of approximately $350 per node.

Sensors and actuators

The *open storm* platform supports an extensive catalog of digital and analog environmental sensors. Typical sensors include (i) ultrasonic and pressure-based water level sensors, (ii) soil moisture sensors, (iii) tipping-bucket and optical rain gages, (iv) automated grab samplers for assessing pollutant loads, and (v) in-situ water quality sensors, including probes for dissolved oxygen, pH, temperature, conductivity, dissolved solids, and oxidation-reduction potential. While many sensors are known to work “out of the box”, new sensors can be quickly integrated by adding device drivers to the sensor node firmware. Support for nearly arbitrary sensors is provided by the microcontroller’s system-on-chip (SoC), which allows for analog and digital peripherals—like analog-to-digital converters, multiplexers, and logic gates—to be generated using programmable blocks in the device firmware. In addition to environmental sensors, the sensor node also includes internal sensors that report device health statistics, including battery voltage, cellular reception strength, and connection attempts. These device health statistics help to diagnose network issues, and can be used as inputs to remote trigger routines. Sensors can be configured remotely using web services (see *cloud services* section). This capability allows users to turn sensors on or off, or to change the sampling frequency of a sensor without reprogramming the device in the field.

The *open storm* platform also supports an array of actuators that can be used to move mechanical devices in the field. These devices are used to guide the behavior of water systems in real-time, by controlling the flow of water in ponds, channels and pipes. Butterfly valves are one common type of actuating device, and are typically used to control discharge from storage elements such as retention basins. Valves can be opened, closed, or configured across any number of partially opened configurations to modulate flows. As with onboard sensors, these devices are operated remotely using commands sent from a server. Control signals can be specified manually, or through automated control algorithms.

Detailed technical information regarding supported sensors and actuators, along with guides for integrating new devices are provided online at `open-storm.org/sensors`.

Cloud services

While sensor nodes can function independently by storing data and making decisions on a local level, integration with cloud services enables system-scale supervision, configuration, and control of field-deployed devices. Like a traditional SCADA system, the cloud services layer facilitates telemetry and storage of sensor data, provides visualization capabilities, and enables remote control of devices—either through manual input or through automated routines. However, unlike a traditional SCADA system, the cloud services layer also allows sensor nodes to communicate with a wide variety of user-defined web applications—including advanced data visualization tools, control algorithms, GIS software, external data ingesters, alert systems, and real-time hydrologic models. By combining real-time supervision and control with domain-specific tools, this architecture enables flexible system-scale control of water assets.

In brief, the cloud services layer performs the following core functions: (1) stores and processes remotely-transmitted data, (2) simplifies management and maintenance of field-deployed sensor nodes, and (3) enables integration with a suite of real-time models, control algorithms, and visualizations. These services are environment-agnostic, meaning that they can be deployed on a local server or a virtual server in the cloud. In practice, however, current *open storm* projects are deployed on popular cloud services—such as Amazon Elastic Compute Cloud (EC2) [81] or Microsoft Azure [82]—to ensure that computational resources flexibly scale with demand. In the following section, we describe the basic architecture, and present example applications that are included with the *open storm* platform.

The cloud services layer follows a simple design pattern, in which applications communicate with sensor nodes through a central database. On the device side, sensor nodes push sensor measurements to the database, and then query the database to determine the latest desired control actions. On the server side, applications query the latest sensor readings from the database, feed these sensor readings into user-defined applications, and then write commands to the database to control the behavior of field hardware remotely. This architecture allows field-deployed sensors to be managed through a single endpoint, and also allows new applications to be developed without modifying critical device firmware.

The database serves dual purposes as both a storage engine for sensor data, and as a communication layer between field-deployed sensors and web applications. The primary purpose of the database is to store incoming measurements from field-deployed sensors. Sensor nodes report measurements directly to the database via a secure web connection—using the same protocol that one might use to access web pages in a browser (HTTPS). The database address (URL) is specified in the sensor node firmware, allowing the user to write data to an endpoint of their choosing. In addition to storing sensor measurements, the database also enables bidirectional communication between the node and cloud-based applications by storing device configuration data, command signals, and data from external sources. Server applications communicate with the sensor node by writing commands to the database. These commands are then downloaded by the sensor node on each wakeup cycle. For example, a real-time control application might adjust outflow from a storage basin by writing a sequence of valve positions to the database. At each sampling interval, the sensor node will query the latest desired valve position and enact the appropriate control action. This system enables bidirectional communication with field-deployed sensor nodes without the need for complex middleware.

For its database backend, the *open storm* project uses InfluxDB, a time-series database that is optimized for high availability and throughput of time-series data [83]. Communications with the database backend are secured through the use of basic authentication (i.e. a username and password), as well as Transport Layer Security encryption (TLS/SSL). The use of basic authentication prevents unauthorized parties from executing malicious commands on the network, while the use of encryption prevents attackers from intercepting sensitive data. Because applications communicate with the sensor node through the database, this means that applications are secured automatically against attackers as well. Altogether, this system comprises a data backend that is secure, maintainable, and extensible.

Applications

The *open storm* platform features a powerful application layer that enables users to process and analyze data, build user interfaces, and control sensor nodes remotely. Applications are implemented by creating a series of subscriptions on the central database. These subscriptions perform one of three actions: (i) *read* from the database, (ii) *write* new entries to the database, and (iii) *trigger* actions based on user-specified conditions. While seemingly simple, this system allows for the development of a wide range of applications. A data visualization platform, for instance, is implemented by continuously querying sensor streams from the database; similarly, automated control is implemented by writing a continuous stream of commands. In the following section, we demonstrate the potential of the open storm application platform by presenting example applications, including a data visualization portal, a push alert system, adaptive control, and real-time integration with hydrologic models.

Network supervision and maintenance tools

Much like a traditional SCADA system, the *open storm* platform provides a web-based graphical user interface for real-time visualization and device configuration. Figure 2.2 shows an example dashboard, with time series of cellular connection strength (top), radial gauges for monitoring battery voltage (center), and real-time depth readings from two sensor nodes (bottom). Time series visualizations are implemented using the Grafana analytics platform [84], which allows users to develop customized dashboards that suit their individual needs. To facilitate remote configuration of sensor nodes, *open storm* also includes a web portal that allows users to change device parameters (such as sampling frequency), control actuator behavior, and set event triggers using a web browser.

Automated alerts and adaptive control

In addition to enabling manual supervision and control, *open storm* also provides a rich interface for triggering automatic actions based on user-specified conditions. Push alerts are one common type of trigger event. Alerts can be used to notify stakeholders of hazardous field conditions, such as flooding, or to recommend control strategies to operators in real time. Alerts are also used to notify the user about the health of the network—for instance, by sending push warnings when node battery voltages drop below a threshold, or by emitting a critical alert when data throughput ceases. These system health alerts allow network outages to be promptly diagnosed and serviced. Alerts can be pushed to a variety of endpoints, including email, text messages, or to social media platforms such as Twitter and Slack [85, 86]. The wide variety of available push notification formats means that the *open storm* platform is suited to handling both (i) confidential alerts for system operators, and (ii) public emergency broadcasts.

In addition to the alert system, subscriptions are also used to implement adaptive sampling and automatic control. Adaptive sampling allows the sampling frequency of the node to be changed remotely in response to weather forecasts, data anomalies, or manual user input [87]. This in turn allows hydrologically interesting events—such as storm events and dam releases—to be measured at an enhanced resolution. To manipulate sampling frequencies in response to changing weather conditions, for instance, weather forecasts are first downloaded into the *open storm* database using an external data ingester. Next, the subscription service parses the incoming data. If the service detects a probability of rain, the sampling frequency of a node is increased. When no precipitation is anticipated, the sampling frequency is decreased, allowing the node to conserve battery power. The same principle is used to implement automated control. The subscription service can be configured as a simple set-point or PID controller, for instance, by computing a control signal based on an input data stream. This controller can in turn be used to optimize outflow from a retention pond, by controlling the position of an outlet valve. More sophisticated control schemes can be implemented by attaching the subscription service to an online model, which optimizes control strategies over an entire stormwater network, achieving system-level benefits. Examples include the MatSWMM and pySWMM software packages [88, 89], which are used to simulate real-time control strategies for urban drainage networks.

Detailed information regarding cloud services and applications can be found at `open-storm.org/cloud`. In addition to the cloud services platform described here, the *open storm* sensor node is also compatible with other cloud-based data management services, such as the CHORDS (Cloud Hosted Real-time Data Services for the Geosciences) portal [90].

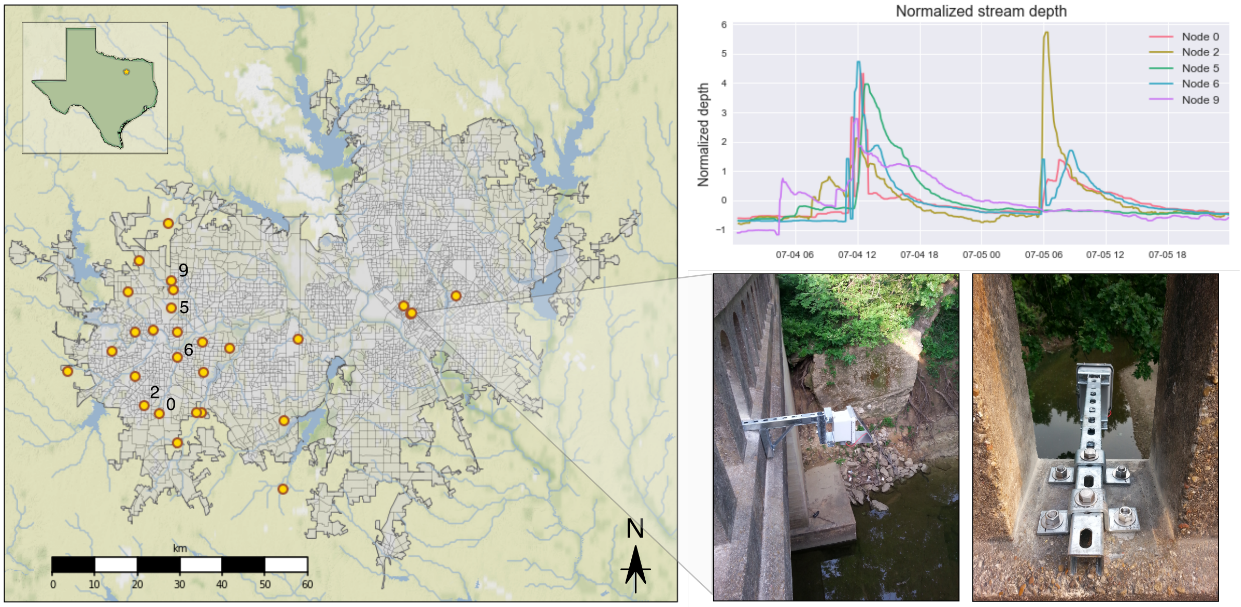


Figure 2.3. Flood monitoring network in the Dallas–Fort Worth metroplex. The map (left) shows current and proposed sensor sites, while the detail photos (bottom-right) show an example bridge-mounted depth sensor node. Time series (top-right) show the response in stream depth to a series of storm events from August 5-6, 2016. From these stage hydrographs, it can be seen that the response varies widely even within a relatively small geographic area.

2.5 Case Studies

To demonstrate the capabilities of the *open storm* platform, we present two ongoing case studies. The first is a real-time flash flood warning network for the Dallas–Fort Worth metroplex in Texas. This deployment detects flash floods at the level of individual roadways, allowing targeted alerts for motorists and improved routing of emergency services during storm events. The second case study is a "smart" stormwater control network in the City of Ann Arbor, Michigan. This deployment aims to improve water quality and mitigate stormwater damage by adaptively timing releases from retention basins across an entire watershed.

Case study 1: Flood monitoring

Located in "flash-flood alley", the Dallas–Fort Worth (DFW) metroplex has historically been one of the most flood-prone areas in the United States [91]. Chronic flooding results in an average of 17 fatalities per year in the state of Texas, with a majority of deaths arising from flash floods [91]. Despite recent efforts to improve stormwater management [92], lack of fine-scale runoff measurements inhibits prediction and communication of flash flood risks. To address this problem, we are using the *open storm* platform to build a real-time flash flood monitoring network. Drawing on the *open storm* real-time alert system, this network aims to improve disaster response by communicating flood risks to emergency managers in real-time, and by generating targeted alerts that will allow motorists to safely navigate around inundated roadways.

To date, urban flash flooding remains a poorly-understood phenomenon. There is currently no model that is capable of generating reliable flash flood estimates in urban areas [6]. Modeling of urban flash floods is complicated by an absence of natural flow paths and interaction of runoff with man-made structures [6]. However, lack of data at appropriate spatial and temporal scales also presents a major challenge. For reliable modeling of flash floods, Berne (2004) recommends using rainfall data at a minimum spatial resolution of 500 meters [28], while a recommended temporal resolution of 1-15 minutes for rainfall is recommended by Smith (2007) [93]. Existing rain gages and river stage monitors are often too sparsely distributed to meet these requirements. Within the DFW metroplex, NWS maintains 12 quality-controlled gages [94], while USGS provides precipitation data at 30 sites [95]. This means that the current spatial resolution of validated rain gages within the DFW metroplex is roughly 1 gage per 600 km^2—too sparse for reliable prediction of flash floods. Likewise, current river stage monitors for the DFW region are largely deployed along mainstems of creeks and rivers with contributing areas ranging from 20 km^2 to 21,000 km^2 (and a median contributing area of 220 km^2). While these gages provide excellent coverage of riverine flooding, they offer limited potential for capturing flash floods.

To fill coverage gaps and enable real-time flash flood forecasting, we are building a wide-area flood monitoring network that is specifically tailored to monitoring flash floods over small-scale catchments (ranging from about 3 to 80 km^2 in size). Our approach is to leverage a large array of inexpensive depth sensors to capture runoff response at the scale of individual roadways, creeks, and culverts. By using inexpensive hardware, we are able to scale our network to a size that would be infeasible with state-of-the-art stage monitoring stations (such as those used by NOAA or USGS). At the time of writing, 40 sensor nodes have been allocated and built for the DFW flood monitoring project, with over 15 nodes currently deployed and reporting. These 40 sensor nodes have been built for a cost of $20,000 USD—less than the cost as a single USGS gaging station [96].[1]

Figure 2.3 presents an overview of the DFW flood monitoring network. The left panel shows a map of the DFW metroplex, with current and proposed sensor node locations. The bottom-right panel shows a detail of a typical sensor node installation. Like most nodes in the network, this node is mounted to a bridge deck with an ultrasonic depth sensor pointed at the stream surface below. The sensor node records the depth to the water surface at a typical time interval of 3-15 minutes. The top-right plot shows a time series of stream depth during two distinct storm events for a sample of nodes on the network. From this plot, it can be seen that the runoff response varies widely between sensor locations, even in a relatively concentrated geographic area. During the second event, for instance, Node 2 (yellow) reports a large increase in discharge, while Node 9 (purple) reports no change in discharge. Comparison of the hydrographs with NEXRAD [97] radar data shows that the variability in stage is largely explained by spatial variability in the rainfall fields[2]. This result confirms the need for increased spatial resolution in stream stage measurements for flash flood monitoring.

The *open storm* platform enables detection and communication of flood risks on spatial and temporal scales appropriate for real-time disaster response and control. Adap-

[1] The installation cost for a USGS stage-discharge streamgaging station is roughly $20,000, with an annual recurring cost of approximately $16,000.

[2] See https://github.com/open-storm/docs.open-storm.org/wiki/Case-study:-Flood-Monitoring-in-Dallas-Fort-Worth

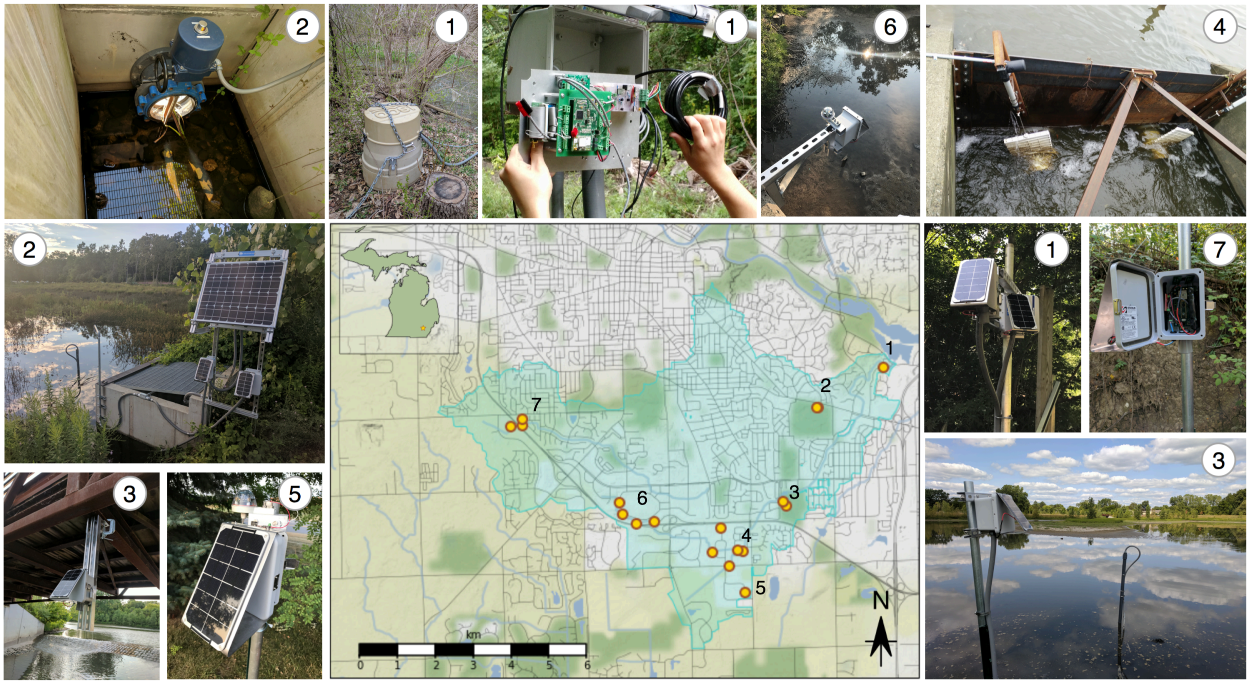


Figure 2.4. The Ann Arbor stormwater control network with selected sites. The blue region of the map shows the boundary of the Malletts Creek watershed, while node locations are indicated by gold markers. The outlet of the watershed (1) is monitored by a USGS gage, along with an automated sampler (1, top-left) and a depth-sensing node (1, center-right). Active control is added through a butterfly valve (2), and dual gate valves (4). The watershed is monitored by an array of depth sensors (6, 7 and 3, bottom-left), soil moisture sensors (5) and in-situ water quality sensors (3, bottom-right).

tive management of traffic during extreme weather events represents one important application of this technology. The Dallas–Fort Worth flood monitoring network could improve disaster response by communicating flood risks to motorists in real-time, thereby allowing them to safely navigate around flooded roadways. This is especially important given that in the US, roughly 74% of fatalities from flooding are motor-vehicle related [1], and in Texas, as much as 93% of flood-related deaths result from walking or driving into floodwaters [91]. Current alert systems are to a large extent insensitive to spatial variability in flood response [75]. However, the *open storm* framework enables targeted alerts that can be integrated into existing mobile navigation apps. In a future that may be characterized by autonomous vehicles and vehicle-to-infrastructure communication [98], this technology could one day be used to adaptively route traffic during extreme weather events.

Case study 2: Controlling Watersheds

As illustrated by the Dallas–Fort Worth flood-monitoring network, real-time measurements can play a pivotal role in providing alerts to stakeholders and improving our understanding of watershed dynamics. However, with the addition of active control, it is possible to not only monitor adverse events, but also to prevent them. The *open storm* platform is capable of enacting control on a watershed scale using distributed valve controllers, adaptive control schemes, and cloud-hosted hydrologic models. Instead of building bigger stormwater systems, operators may use real-time control to make better use of existing water infrastructure, mitigate flooding, and decrease contaminant loads into sensitive ecosystems.

The *open storm* framework is presently being used to control an urban watershed in the City of Ann Arbor, Michigan. The *Malletts Creek* watershed—a 26.7 km^2 tributary of the Huron River—has traditionally served as a major focal point in the city's strategy to combat flooding and reduce runoff-driven water quality impairments [99]. Given its proximity to the Great Lakes, water resource managers have placed an emphasis on reducing nutrient loads from urban runoff. A majority of the discharge in Malletts creek originates from the predominantly impervious upstream (southwestern) reach of the watershed, while a significant, but smaller portion of the discharge originates from the central reach of the watershed. For this reason, local water resource managers have constructed a number of flood-control basins in the upstream segments of the catchment. It is these basins that are now modified to allow for real-time control of the watershed.

The watershed is modified for control at two locations by retrofitting existing basin outlets with remotely-operated valves (Figure 2.4). The first control point is a stormwater retention pond in the southern part of the watershed (shown in red in Figure 2.5). While originally designed as a flow-through (detention) pond, the addition of two 30 cm diameter gate valves allows for an additional 19 million liters of water to be actively retained or released. The second control point is a smaller retention pond, located in the central reach of the watershed (shown in green in Figure 2.5). This control site is retrofitted with a rugged 30 cm diameter butterfly valve. The position of each valve is controlled via an attached sensor node, which relays commands from a remote server.

Each sensor node is equipped with a pair of ultrasonic sensors: one to measure the water depth at the pond, and one to measure the depth of the outflow stream. The control sites operate entirely on 12V battery power, along with a solar panel to recharge the battery during daylight hours. This configuration allows the controller to remain in the field permanently, without the need for a dedicated external electricity source.[3]

In addition to the two control sites, the Ann Arbor network is also instrumented with more than twenty sensor nodes that monitor system performance and characterize real-time site conditions. Using a combination of ultrasonic depth sensors, optical rain gages, and soil conductivity sensors, these nodes report stream stage, soil moisture, soil temperature, and precipitation accumulation approximately once every 10 minutes (with an increased resolution of 2-3 minutes during storms). An additional set of nodes is deployed to measure water quality—including dissolved oxygen, pH, temperature, oxidation reduction potential, conductivity, temperature—as well as an automated grab sampler for capturing contaminants of interest (such as heavy metals and microbes). These nodes are deployed at the inlet and outlet of constructed wetlands to determine how real-time control affects the removal of pollutants.

Measurements from the sensor network are validated using an external United States Geological Survey flow measurement station (USGS station 4174518), located at the watershed outlet. These federally-certified measurements are available freely on the web, making them relatively easy to ingest into the *open storm* framework as an external data source. Furthermore, localized weather forecasts are ingested from public forecasting services (darksky.net) to provide daily, hourly, and minute-level forecasts to inform the control of each site in the network [100]. These external data sources allow for near-instant validation of sensor data, and provide a holistic "snapshot" of system states.

We confirm the effectiveness of the control network through a simple experiment. In this experiment, stormwater is retained at an upstream control site, then released gradually to maximize sedimentation and reduce erosion downstream. While it is known that the addition of control valves affords many localized benefits—such as the ability to increase retention and capture sediments [101]—the goal of this experiment is to test the extent to which control of individual sites can improve watershed-scale outcomes. The

[3] With two people, installation at each site takes approximately one day. This includes time dedicated to mounting valves, sensors, and remotely-testing the equipment.

control experiment takes place on a river reach that stretches across three sites: a retention pond (upstream), a constructed wetland (center), and the watershed outlet. Figure 2.5 (right) shows the three test sites within the watershed, with the fractional contributing area of each site indicated by color. In this system, runoff flows from the retention pond (red) to the watershed outlet (blue) by way of an end-of-line constructed wetland (green) designed to treat water, capture sediments, and limit downstream erosion. Erosion, in particular, has been shown to be primary source of phosphorus in the watershed [87], thus emphasizing the need to reduce flashy flows. While the wetland serves a valuable purpose in improving water quality, it is sized for relatively small events. Specifically, the basin is designed to hold up to 57 million liters of stormwater but experiences as much as 760 million liters during a ten-year storm. Thus, it often overflows during storms, meaning that treatment benefits are bypassed. To maximize treatment capacity, a sensor node is placed into the wetland to measure the local water level and determine the optimal time to release from the retention pond upstream.

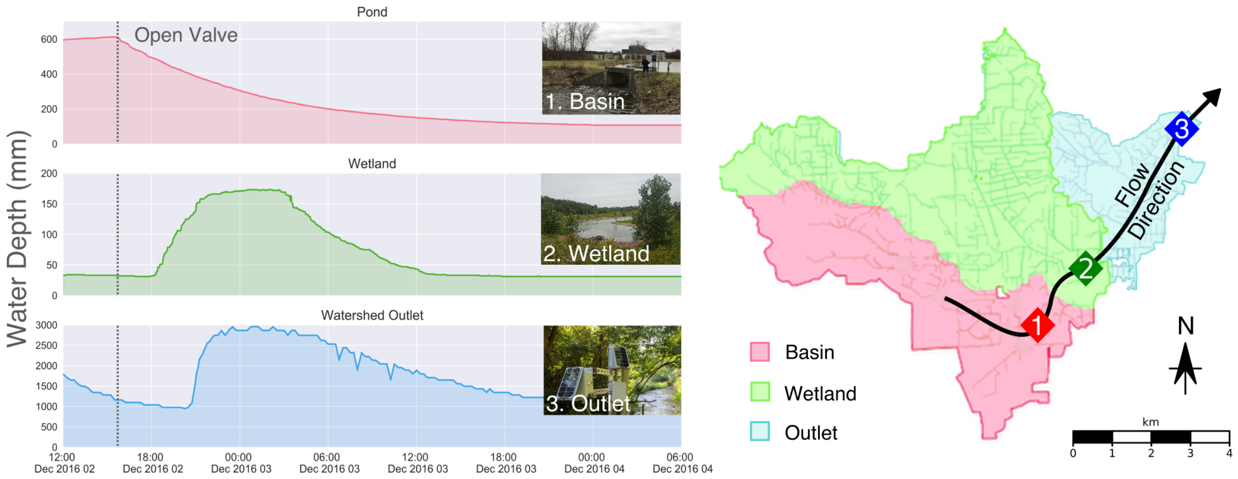


Figure 2.5. Malletts Creek control experiment. The left panel shows time series of water depth from 12:00 pm on December 2 to 6:00 am on December 4, 2016. The right panel shows the location of the three sites in the watershed, with the partitioned contributing areas of each location corresponding to the colors of the time series plots.

At the outset of the experiment, water is held in the upstream retention pond following a storm on December 1, 2016. Residual discharge from the original storm event can be observed as a falling hydrograph limb at the USGS gaging station (blue) during the first 10 hours of the experiment (Figure 2.5). The sensor located at the wetland is used to determine the time at which it is safe to release upstream flows without overflowing the wetland (Figure 2.5). Water is initially released from the pond at 4:00 pm on December 2, as indicated by a drop in the water level of the pond. Two hours later, the water level in the wetland begins to rise due to the discharge arriving from upstream. Finally, after another three hours, the discharge wave reaches the outlet, where it is detected by the USGS flow station. Over the course of the controlled release, the station registers roughly 19 million liters of cumulative discharge.

The control experiment shows demonstrable improvements in system performance compared to the uncontrolled case. While the water quality benefits will be measured in the coming year, a number of likely benefits can be posited. As measured, over 19 million liters were removed from the storm window and retained in the basin following the storm event. The residence time of the water in the pond increased by nearly 48 hours, increasing the potential for sedimentation [101]. The removal of stormwater flows also resulted in attenuation of the downstream hydrograph. The peak flows at the watershed outlet were measured to be 0.28 m^3/s during the storm, but would have been nearly 0.60 m^3/s had the valves in the basin not been closed. Based on prior studies in the watershed—which showed that flows in the stream correlate closely with suspended sediment concentrations—it can be estimated that the flows from the basin were discharged at roughly 60 mg/L, rather than 110 mg/L, thus nearly halving the concentration of suspended solids and total phosphorus in the flows originating from the controlled basin [87]. Moreover, the controlled experiment enhanced the effective treatment capacity at the wetland downstream, which would have overflowed during the storm, thus not treating the flows from the upstream pond. As such, the simple addition of one upstream valve provided additive benefits across a long chain of water assets, demonstrating firsthand how system-level benefits can be achieved beyond the scale of individual sites. While the water quality impacts of active control deserve further assessment, this study opens the door for adaptive stormwater control at the watershed scale. Rather than optimizing the performance of isolated sites, the *open storm* platform can be used to determine the optimal control strategy for an entire watershed, then enact it in real-time.

2.6 Conclusion

Open storm is an all-in-one, "batteries included" platform for monitoring and managing urban water systems. Its emphasis on extensive configurability, real-time response, and automated control make it an ideal choice for water system managers and environmental researchers alike. While many open hardware platforms exist, *open storm* is the first open-source, end-to-end platform that combines sensing, control and cloud computing in service of water resources management. Aside from providing a technological blueprint, *open storm* addresses the real-world requirements that can be expected in water resources applications, such as field-robustness, low-power operation and system-scale coordination. The *open storm* project has shown proven results in extending the capabilities of existing stormwater systems: both by increasing the spatiotemporal resolution of measurements, and by actively improving water quality through real-time control. However, *open storm* is not just a platform—it's also a community of researchers, stakeholders and decision-makers who are dedicated to realizing smarter water systems. To assist in the dissemination and development of smart water systems, we are creating a living document at `open-storm.org` in order to share standards, reference materials, architectures, use cases, evaluation metrics, and other helpful resources. We invite users to participate in this project by sharing their experiences with designing, deploying and maintaining smart water systems.

Published version

This chapter was reprinted with permission from the article [102]:
Bartos M, Wong B, Kerkez B. (2018) *Environ. Sci.: Water Res. Technol.*, **4**,346-358

Chapter 3

Windshield Wipers on Connected Vehicles Produce High-Accuracy Rainfall Maps

Abstract

Connected vehicles are poised to transform the field of environmental sensing by enabling acquisition of scientific data at unprecedented scales. Drawing on a real-world dataset collected from almost 70 connected vehicles, this study generates improved rainfall estimates by combining weather radar with windshield wiper observations. Existing methods for measuring precipitation are subject to spatial and temporal uncertainties that compromise high-precision applications like flash flood forecasting. Windshield wiper measurements from connected vehicles correct these uncertainties by providing precise information about the timing and location of rainfall. Using co-located vehicle dashboard camera footage, we find that wiper measurements are a stronger predictor of binary rainfall state than traditional stationary gages or radar-based measurements. We introduce a Bayesian filtering framework that generates improved rainfall estimates by updating radar rainfall fields with windshield wiper observations. We find that the resulting rainfall field estimate captures rainfall events that would otherwise be missed by conventional measurements. We discuss how these enhanced rainfall maps can be used to improve flood warnings and facilitate real-time operation of stormwater infrastructure.

3.1 Introduction

Accurate rainfall measurements are essential for the effective management of water resources [27]. Historical rainfall records are used extensively in the design of water infrastructure [103], while at finer scales, real-time rainfall measurements are an integral component of flood forecasting systems [6]. Despite the central role that precipitation measurements play in the design and operation of water infrastructure, current methods for measuring precipitation often do not provide the spatial resolution or measurement certainty required for real-time applications [6]. As the demand for real-time precipitation data increases, new sensing modalities are needed to address deficiencies found in conventional data sources.

The need for high-resolution precipitation estimates is perhaps best illustrated by the problem of urban flash flooding. Flooding is the number one cause of natural disaster fatalities worldwide, with flash floods accounting for a majority of flooding deaths in developed countries [1]. Despite the risks posed by flash flooding, there is "no existing model [that is] capable of making reliable flash flood forecasts in urban watersheds" [6]. Flash flood forecasting is to a large extent hindered by a lack of high-resolution precipitation data, with spatial resolutions of < 500 m and temporal resolutions of 1-15 minutes required for urban areas [28, 93].

Contemporary rain measurement technologies—such as stationary rain gages and weather radar—struggle to achieve the level of precision necessary for flash flood forecasting. While rain gages have long served as a trusted source of surface-level precipitation measurements [104], they often fail to capture the spatial variability of rain events, especially during convective storms [29–31]. This inability to resolve spatial patterns in rainfall is made worse by the fact that the number of rain gages worldwide is rapidly declining [27]. Weather radar is a useful tool for capturing the spatial distribution of rainfall. However, radar-rainfall estimates are subject to large spatial and temporal uncertainties [33–35, 105]. Additionally, weather radar tends to show systematically large biases for major flood events, and may perform poorly for small watersheds [93], making urban flood forecasting problematic.

The rise of connected and autonomous vehicles offers an unprecedented opportunity to enhance the density of environmental measurements [37, 38]. While dedicated sensor networks are expensive to deploy and maintain, fleets of connected vehicles can capture real-time data at fine spatial and temporal scales through the use of incidental onboard sensors. With regard to rainfall measurement, windshield wiper activity offers a novel means to detect the location and timing of rainfall with enhanced precision. When used in conjunction with modern signal processing techniques, wiper-based sensing offers several attractive properties: (i) vehicles achieve vastly improved coverage of urban areas, where flood monitoring is important; (ii) windshield wiper intensity is easy to measure and requires little overhead for processing (as opposed to video or audio data); and (iii) vehicle-based sensing can be readily scaled as vehicle-to-infrastructure communication becomes more widespread. Moreover, many new vehicles come equipped with optical rain sensors that enable direct measurement of rainfall intensities. When paired with data assimilation techniques, these sensors may enable even higher-accuracy estimation of rainfall fields compared to wipers alone.

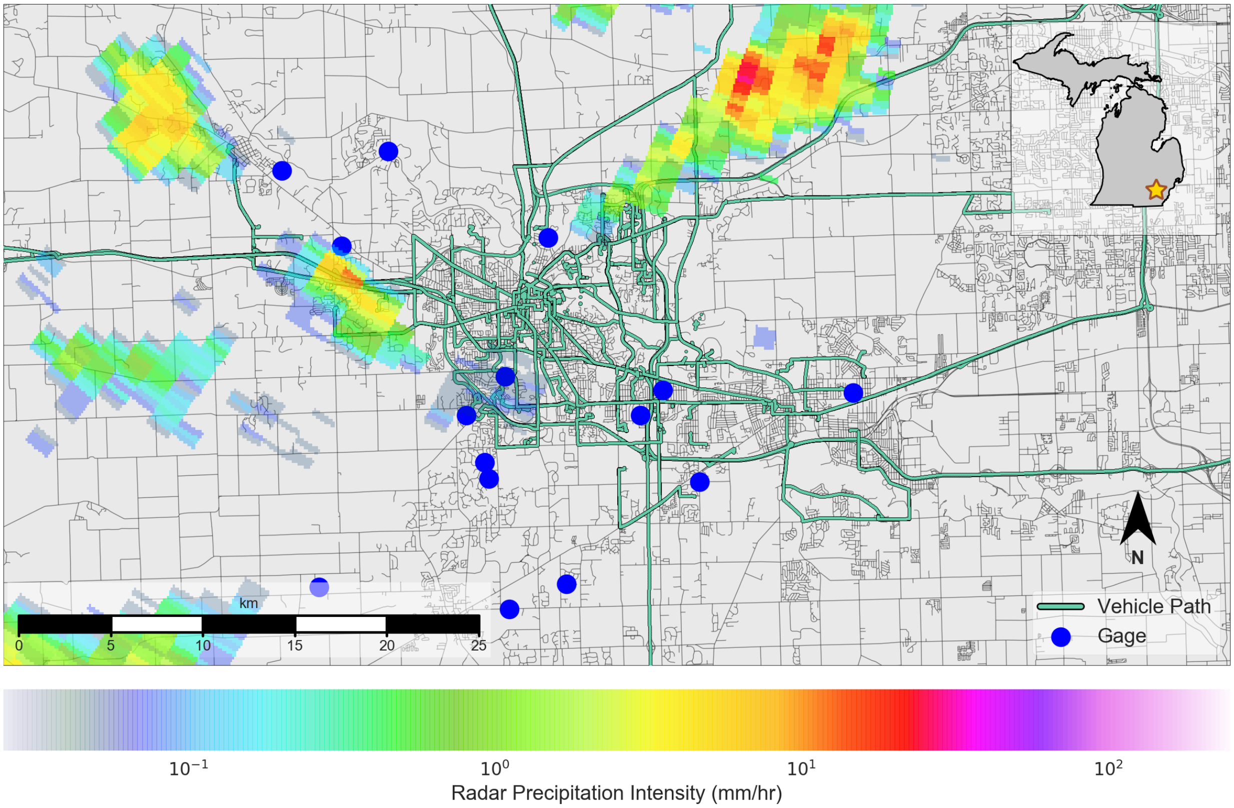


Figure 3.1. Overview of vehicle study area. Map depicts vehicle and rainfall conditions on June 12, 2014. Blue circles represent rain gages. Vehicle paths are shown as green lines, while roads are shown in gray. A radar overlay shows the average precipitation intensity as estimated by radar.

While a small number of studies have investigated vehicle-based precipitation measurements, the results of these studies are strictly based on simulated wiper data instead of real measurements. As such, the premise that windshield wiper data can be used to improve rainfall estimates has never been verified using a large real-world dataset. Hill (2015) combines simulated binary (wet/dry) rainfall sensors with weather radar observations to generate improved areal rainfall estimates, which are then validated against rainfall fields produced by interpolation of tipping-bucket rain gages [37]. Similarly, Haberlandt (2010) combines simulated vehicle wiper measurements with rain gage observations to improve rainfall field estimates, and then validates the resulting product against weather radar [38]. Although these studies highlight the potential for vehicle-based measurements to improve the spatial and temporal resolution of rainfall estimates, their findings have not yet been validated using data from real-world connected vehicles.

To address these challenges, this study leverages windshield wiper measurements collected from nearly 70 vehicles to produce corrected rainfall maps (see Figure 3.1 for a description of the study area and data sources). In the first part of this paper, we demonstrate that windshield wiper measurements offer a reliable indicator of rainfall by comparing wiper measurements against dashboard camera footage that indicates the ground truth binary rainfall state (raining/not raining). In the second part of this paper, we develop a Bayesian data fusion procedure that combines weather radar with vehicle-based wiper measurements to produce an updated probabilistic rainfall field map. We validate this novel data product by showing that it is more effective than the original radar data at predicting the binary rainfall state. Finally, we discuss how these enhanced rainfall maps can be used to improve flood warnings and facilitate real-time operation of stormwater infrastructure.

3.2 Results

Windshield wipers improve binary rainfall detection

Windshield wiper measurements enhance rainfall estimation by enabling greater certainty about the timing and location of rainfall. While wiper intensity on its own is generally a poor predictor of rainfall intensity (see Figure S3.1 in the Supplementary Informa-

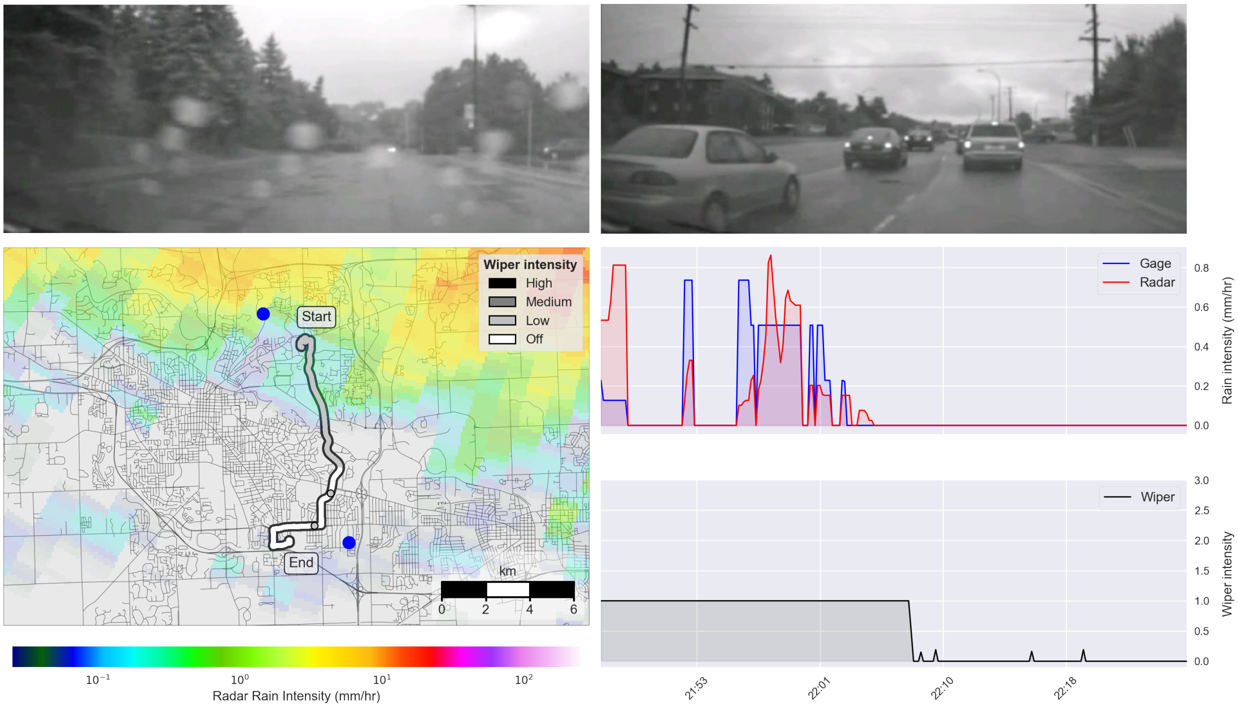


Figure 3.2. Analysis of a vehicle trip. The vehicle trip occurred from 21:46 - 22:26 on August 11, 2014. The top two panels show video footage during the rainy (left) and dry (right) segments of the trip. The bottom left panel shows a map of the vehicle's trip, with the wiper intensity indicated by color. A radar overlay shows the average rainfall intensity over the 40-minute time period. Blue circles represent the gages nearest to the vehicle path. The two bottom right panels show the precipitation intensity as estimated by radar and gage measurements (center), and the 1-minute average wiper intensity (bottom). Photographs are reproduced with permission from the University of Michigan Transportation Research Institute.

tion), we find that wiper status (on/off) is a stronger predictor of binary rainfall state than either radar or gage-based measurements. This result suggests that vehicle-based measurements can be used to validate and correct rainfall fields derived from conventional data sources.

Wiper measurements provide a more accurate indicator of binary rainfall state than either radar or gage measurements. We determine the binary classification performance for each technology (gages, radar and wipers) by comparing the measured rainfall state with co-located dashboard video footage. Dashboard video is taken to represent the ground truth, given that the presence or absence of rainfall can readily be determined by visually inspecting the windshields for raindrops. Figure 3.2 shows an example of co-located radar, gage, wiper and camera measurements for a single vehicle trip. The top two frames show dashboard camera footage collected over the course of the vehicle trip. Rainfall is visible during the first half of the trip (top left) while no rain can be seen during the second half of the trip (top right). The map (bottom left) shows the path of the vehicle along with (i) the reported wiper intensity, (ii) the average radar rainfall intensity during the trip, and (iii) the two nearest rain gages. Two time series (right) compare radar and gage measurements of rainfall intensity near the vehicle's location (center right) with reported wiper intensity (bottom right). The binary classification performance for each data source is assessed by manually labeling the ground truth rainfall state based on the dashboard camera footage, and then comparing these labels with the binary rainfall state predicted by co-located wiper, radar and rain gage data sources.

Table 3.1. Classification performance of each rainfall measurement technology. The true positive rate indicates the percentage of instances where the given technology successfully detects rainfall when rainfall is actually occurring. The true negative rate indicates the percentage of instances where the technology does not detect rainfall when rainfall is not occurring.

Metric	Gage	Radar	Wiper
True Positive Rate (%)	44.5	89.5	93.1
True Negative Rate (%)	96.7	97.5	98.2

Comparing radar, gage, and wiper measurements with co-located vehicle footage across three storm events, we find that wiper status is the best estimator of binary rainfall state, with a true positive rate (TPR) of 93.1%, and a true negative rate (TNR) of 98.2%. By comparison, weather radar achieves a smaller TPR of 89.5%, while stationary gages show a much smaller TPR of 44.5% (see Table 3.1). These results can partly be explained by the superior spatial and temporal resolution of the wiper measurements. Wipers detect intermittent changes in rainfall at a temporal resolution on the order of seconds, while radar and gage measurements can only detect the average rate over a 5-minute period. When ground truth camera observations are collected at a 3-second temporal resolution, the benefit of wiper measurements over radar measurements becomes even more pronounced, with a TPR advantage of 5.2%, a TNR advantage of 7.7%, and an overall wiper TPR of 97.0% (see the supplementary note on factors affecting binary detection performance). The results of this analysis suggest that conventional rainfall measurement technologies can be enhanced through the inclusion of vehicle-based measurements.

Assimilation of wiper data yields corrected rainfall maps

Based on the observation that wiper measurements are a strong binary predictor of rainfall, we develop a Bayesian filtering framework that combines radar rainfall estimates with wiper observations to generate corrected rainfall maps. Radar is first used to estimate a prior distribution of rainfall intensities. This prior is then updated with wiper observations to produce a corrected rainfall intensity field that better captures the binary rainfall state. The results of this filtering procedure are demonstrated in Figure 3.3, which shows the original rainfall intensity field (top) along with the corrected rainfall intensity field (bottom). Vehicle paths are shown (bottom) to highlight the effect of wiper measurements on the posterior rainfall intensity distribution. In cases where both radar and wipers agree on the binary rainfall state, the rainfall intensity field remains unchanged. For example, when the wiper and radar intensities are both nonzero (as seen in the bottom-left panel, leftmost vehicle), the posterior rainfall intensity is simply equal to the prior rainfall intensity. In other cases, vehicles detect no rainfall in regions where radar had previously estimated rainfall (bottom-left panel, rightmost vehicle). In these cases, the Bayesian filter reduces the intensity of the rainfall field within the proximity of the vehicle. Conversely, in the case where vehicles detect rainfall in regions where little to no rainfall was observed in the original dataset (right panel), the Bayesian filter amplifies the rainfall intensity field within the vicinity of the vehicle, resulting in a new rainfall intensity distribution that better represents the binary rainfall state. The predicted rainfall intensity depends on both the wiper measurement and the intensity of the radar rainfall prior within the neighborhood of the vehicle. Thus, vehicles located near a prior rainfall front (bottom-right panel, center of frame) will have a larger effect on the posterior rainfall intensity than vehicles located far away from a prior rainfall front (bottom-right panel, top of frame). For a more detailed view of the evolution of the rainfall field under both the original and corrected data sets, refer to Video S1.

The wiper-corrected rainfall field predicts the binary rainfall state with greater accuracy than the radar-only data product. To validate the wiper-corrected rainfall field, we use an iterated “leave-one-out” approach, in which an updated rainfall field is generated while excluding a vehicle, and the resulting data product is compared against the measured rainfall state of the omitted vehicle. Repeating this process for each vehicle yields the receiver operator characteristics shown in Figure 3.4. These curves map the relationship between the TPR and TNR for both the original rainfall field (radar only) and the corrected rainfall field (radar and wiper). Curves located closer to the upper-left corner (i.e. those

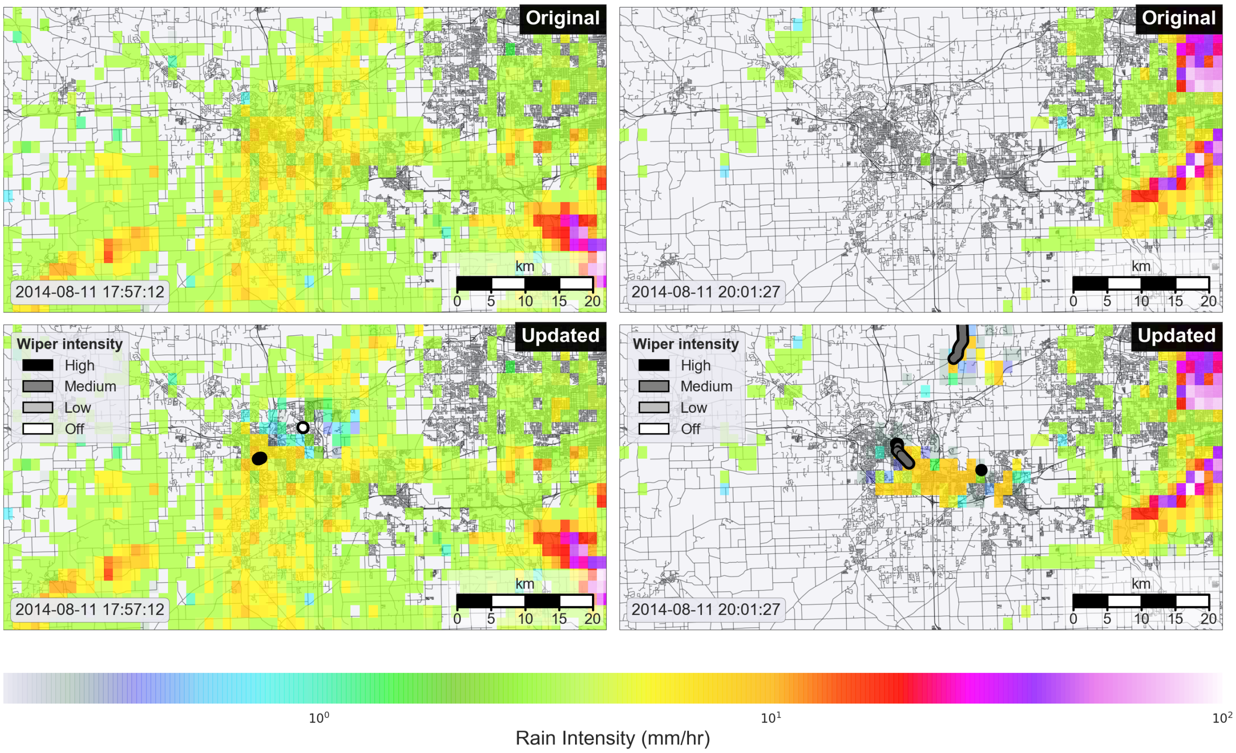


Figure 3.3. Original and updated rainfall maps. Top (left and right): Original prior weather radar rainfall intensity map. Radial radar scans have been resampled to a 1 km grid to ensure computational tractability. Bottom (left and right): updated posterior rainfall intensity map, combining radar data with wiper measurements using the Bayesian filter. In the bottom left panel, a "hole" in the rainfall field occurs when a vehicle detects no rain in a location where radar alone estimated rain. In the bottom right panel, vehicles detect rainfall where radar previously did not detect rainfall.

with a larger area under the curve) exhibit the best performance, given that they have a large true positive rate and a small false negative rate. Based on these curves, it can be seen that the corrected data product performs consistently better than the original radar product at predicting the presence or absence of rain, with a TPR and TNR close to unity. The overall performance of the updated rainfall product—as measured by the area under the curve (AUC)—is roughly 0.957, compared to only 0.878 for the original radar data. These results confirm that inclusion of vehicle-based measurements enables improved prediction of the underlying rainfall field.

3.3 Discussion

The enhanced rainfall maps developed in this study have the potential to assist in the real-time operation of transportation and water infrastructure. In particular, high accuracy rainfall field estimates will enable improved prediction of flash floods in urban

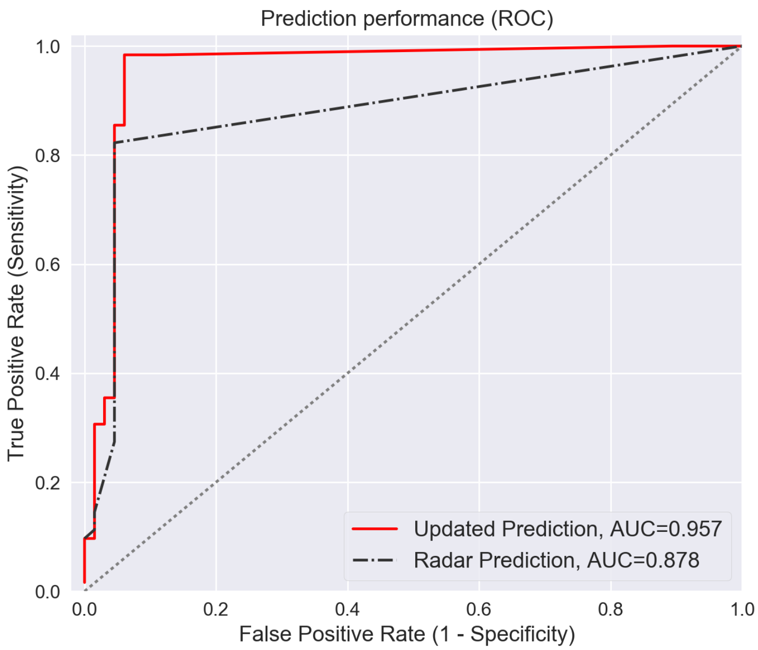


Figure 3.4. Binary classification performance of updated rainfall product. Receiver operator characteristic (ROC) curves indicate the rainfall state prediction accuracy for the original radar estimate and the updated (wiper-corrected) data product. The area under the curve (AUC) measures overall classification performance.

centers, and will help to inform real-time control strategies for stormwater systems. As mentioned previously, flash flood forecasting is contingent on high-resolution areal rainfall estimates, with accurate measurements on the order of 500 m or finer required for forecasting in urban areas. By enabling real-time validation and filtering of radar rainfall estimates, vehicle-based sensors may help fill measurement gaps and improve the prediction of flood events near roadways. Monitoring of roadways is especially important given that in the US, roughly 74% of flood fatalities are vehicle related [1]. As connected and autonomous vehicles become more widely adopted, the spatial coverage and measurement certainty of this new sensing modality will be even further enhanced.

In addition to assisting with flash flood response, high-precision rainfall data products may one day inform the operation of new "smart" water infrastructure. Recent work has highlighted the potential of "smart" water systems to mitigate water hazards through real-time control of distributed gates, valves and pumps [19, 59, 60, 87, 102]. When informed by accurate and timely data, these systems can significantly reduce operating costs, prevent combined sewer overflows, and halt the degradation of aquatic ecosystems by adaptively reconfiguring water infrastructure in real time [19, 102]. However, recent findings suggest that optimal control strategies for "smart" water systems are highly sensitive to the location, timing and intensity of rainfall inputs [106]. In this regard, the wiper-corrected rainfall product presented in this study may help to enable more fine-grained control of water infrastructure by reducing uncertainty in conventional rainfall field estimates.

While this work evaluates the updated rainfall product in terms of its ability to predict the binary rainfall state, future work should use vehicle-based sensors to further validate and improve the predicted rainfall intensity. Currently, visual inspection of the ground truth data source (camera footage) only allows for verification of the binary rainfall state and not the predicted rainfall intensity. Other potential sources of ground truth rainfall intensity, such as stationary rain gages, are also problematic. While rain gages provide an independent source of rainfall intensity data, they are only able to produce estimates of rainfall accumulation at point locations every 5 minutes, and are often far removed from the nearest vehicle path. Moreover, as shown in Table 3.1, gages are by far the poorest predictor of the binary rainfall state among all data sources considered. These issues raise questions as to the appropriateness of rain gages as a source of ground truth rainfall intensity data. With these issues in mind, a natural extension of the work presented in this paper could use other vehicle-based sensors to better estimate the rainfall intensity at each vehicle's location. Drawing on dashboard camera footage, object detection techniques could be used to isolate and count raindrops on the windshield of each vehicle. The volume of rainfall deposited over each wiper interval may then be estimated, thereby yielding an estimate of rainfall intensity at the vehicle's location. Similarly, many newer vehicles feature optical rain sensors that are capable of measuring precipitation rate directly. When combined with the Bayesian sensor fusion framework presented in this study, these sensors could enhance the accuracy of the estimated rainfall intensity field. While outside the scope of this work, these techniques represent promising directions for future research and should be considered in subsequent studies.

3.4 Conclusions

This study generates enhanced probabilistic rainfall maps by combining conventional radar-based precipitation fields with ubiquitous windshield wiper measurements from almost 70 unique vehicles. We find that while windshield wiper intensity is a poor predictor of rainfall intensity, wiper activity is a stronger predictor of binary rainfall state than conventional radar and gage-based data sources. With this result in mind, we develop a novel Bayesian filtering framework that combines a radar-based rainfall prior with binary windshield wiper observations to produce an updated rainfall map. We find that the Bayesian filtering process is effective at detecting changes in the rainfall field that conventional measurement technologies may otherwise miss. We validate the updated rainfall data product by assessing its ability to reproduce the binary rainfall state anticipated by an omitted vehicle. Based on this analysis, we find that the corrected rainfall field is better at predicting the binary rainfall state than the original radar product. As connected vehicles become more widespread, the ubiquitous sensing approach proposed by this study may one day help to inform real-time warning and control systems for water infrastructure by providing fine-grained estimates of the rainfall field.

3.5 Materials and Methods

Evaluating vehicle-based measurements

In the first part of this study, we assess the degree to which windshield wiper activity serves as a proxy for both rainfall intensity and binary rainfall state. First, wiper measurements are compared against conventional rainfall measurement technologies to determine if there is a direct relationship between wiper intensity and rain intensity. Next, we assess the degree to which each data source reflects the ground truth rainfall state by comparing measurements from all three sources (gages, radar and wipers) with vehicle-based video footage. Video footage provides instantaneous visual confirmation of the rainfall state (raining or not raining), and is thus taken to represent the ground truth. We characterize the binary classification performance of each technology in terms of its true positive and true negative rates.

To ensure that our analysis is computationally tractable, we isolate the study to a subset of three storms in 2014. We assess the validity of our procedure for storms of different magnitudes by selecting a large storm (2014-08-11), a medium-sized storm (2014-06-28) and a small storm (2014-06-12). Storms are selected during the summertime months to avoid conflating rainfall measurements with snow measurements. The year 2014 is chosen because it is the year for which the greatest number of vehicles are available. Unless otherwise specified, data are co-located using a nearest neighbor search. For comparison of wiper and gage readings, we select only those gages within a 2 km range of any given vehicle.

Data sources

We consider four data sources: (i) stationary rain gages, (ii) weather surveillance radar, (iii) vehicle windshield wiper data, and (iv) vehicle dashboard camera footage. We provide a brief description of each data source here:

Gage data are obtained from personal weather stations maintained by the Weather Underground [107]. Within the city of Ann Arbor (Michigan), Weather Underground hosts 21 personal weather stations, each of which yield rainfall estimates at a time interval of approximately 5 minutes. Locations of gages are indicated by blue circles in Figure 3.1. Although verified gage data from the National Weather Service (NWS) and the National Oceanic and Atmospheric Administration (NOAA) are available, Weather Underground gages are selected because (i) NOAA and NWS each maintain only a single gage in the city of Ann Arbor, meaning that intra-urban spatial variations in precipitation intensity cannot be captured, and (ii) the temporal resolution of NOAA and NWS gages are relatively coarse for real-time applications (with NOAA offering a maximum temporal resolution of 15 minutes and NWS offering a maximum temporal resolution of 1 hour).

Weather radar observations are obtained from NOAA's NEXRAD Level 3 Radar product archive [108]. We use the "Instantaneous Precipitation Rate" data product (listed as variable code 176 in the NEXRAD Level 3 archive [108]). Radar precipitation estimates are obtained at a temporal resolution of 5 minutes, and a spatial resolution of 0.25 km by 0.5 degree (azimuth). Radar station KDTX in Detroit is used because it is the closest radar station to the City of Ann Arbor. Radial radar scans are interpolated to cartesian coordinates using a nearest neighbor approach.

Vehicle-based wiper intensities are obtained from the University of Michigan Transportation Research Institute (UMTRI) Safety Pilot Model Deployment database [109]. For each vehicle, this dataset includes time series of latitude, longitude, and windshield wiper intensity at a temporal resolution of 2 milliseconds. Windshield wiper intensity is given on an ordinal scale from 0 to 3, with 0 indicating that the wiper is turned off, 1 representing the lowest wiper intensity, and 3 representing the highest wiper intensity. A wiper reading of 4 indicates that the vehicle's "mister" is activated, distinguishing between wiper use for rain removal and wiper use for windshield cleaning. For this study, wiper usage for cleaning (i.e. wiper mode 4) was filtered out before the analysis. Note that wiper intensity codes are based on electrical signals generated by the wiper itself, meaning that no manual wiper mode classification is needed. For the year 2014, 69 unique vehicles are available in the UMTRI dataset. However, typically less than ten vehicles are active at any given time during the observation period. Vehicles with no sensor output or invalid readings were removed from the dataset prior to the analysis (see the Supplementary Note for more details). Other sources of human error (such as accidentally turning wipers on), are captured by the true positive and true negative rates included in Tables 3.1 and S3.1.

Camera observations are also obtained from the UMTRI vehicle database [109]. Located on the inside of each vehicle, cameras provide streaming video footage of the windshield, side-facing windows, rear-facing windows, and the driver. For the purposes of validation, we use the front-facing windshield camera. Camera frames are manually inspected for rain drops striking the windshield. Time intervals where rain is observed are classified as "raining"; similarly time intervals where no new droplets are observed are classified as "not raining". Manual inspection and labeling of the video data was performed independently by two reviewers to ensure robustness.

A Bayesian filtering framework

In the second part of this study, we develop a Bayesian filtering framework that combines binary wiper observations with radar-based rainfall intensity measurements to generate corrected rainfall maps. In simple terms, the Bayesian filter generates an updated rainfall field, in which binary (on/off) wiper measurements adaptively correct the underlying radar rainfall field. Windshield wiper status is taken to represent a measurement of the ground truth binary rainfall state, given that it is a better predictor of the binary rainfall state than radar- or gage-based measurements. Under this framework, four distinct cases are possible. If both the wiper and radar measure precipitation, the radar reading is taken to be correct, and the original rainfall field remains the same. Similarly, if neither the wiper nor the radar measure precipitation, the radar rainfall field remains zero. However, if the radar measures precipitation at a target location and the wiper does not, then the filter will update the rainfall field such that rainfall intensity is reduced within the proximity of the vehicle (with a decay pattern corresponding to the Gaussian kernel and an intensity of zero at the location of the wiper reading). Similarly, if the wiper measures precipitation, but the radar measures no precipitation, the rainfall intensity will be increased within the proximity of the vehicle (by combining the local distribution of the radar rainfall prior with a point estimate of rainfall intensity based on the wiper intensity). In our implementation, provided that no other information is available, this point estimate is generated using the empirical rainfall intensity distribution associated with the given wiper intensity. The empirical rainfall intensity distributions associated with each wiper intensity are shown in Figure S3.1 in the Supplementary Information.

Note that while wiper intensity by itself does not exhibit a strong correlation with rainfall intensity, the Bayesian filter uses both wiper and radar measurements to generate the posterior rainfall intensity estimate. In other words, the posterior rainfall intensity at the vehicle's location is a probabilistic estimate that depends on both the wiper-based estimate and the local prior intensity within the neighborhood of the vehicle. Thus, a nonzero wiper measurement located far away from a radar rainfall front will result in a smaller posterior intensity than one located near a radar rainfall front (as discussed in the results section and shown in Figure 3.3). The relative contribution of the wiper measurement and radar prior are controlled using a weighting parameter representing the user's trust in each data source. This probabilistic assimilation of data sources helps to reduce the uncertainty associated with using the wiper intensity to estimate rainfall intensity. It should be noted that other methods for obtaining a point estimate of rainfall intensity are possible—such as choosing the closest nonzero intensity in the radar rainfall prior. For newer vehicles equipped with rain sensors, the rainfall intensity can also be measured directly using the sensor output. As mentioned in the discussion section, however, it is currently difficult to evaluate the relative accuracy of these approaches, given the lack of reliable ground truth rainfall intensity data at the appropriate spatial and temporal scales.

A more formal description of the filtering framework is given here in terms of a noisy sensor model (for additional details, see [110]). Consider a noisy sensor model in which each sensor produces a binary measurement given a target state. The target state is represented as a random tuple $\boldsymbol{z} = (q, \boldsymbol{I})$ where q is a location state (e.g. the latitude and longitude at the target), and $\boldsymbol{I}$ is an information state (e.g. the precipitation intensity at the target) with all the random quantities indicated by bold italics. We denote by M_t the event that sensors correctly measure the intensity, and by $\overline{M}_t$ the event that sensors fail to measure the intensity correctly. The joint measurement likelihood at any time t is given by:

$$p(M_t|\boldsymbol{z}, x_t) \tag{3.1}$$

where x_t represents the locations of the sensors at time t. Equation 6.2 yields the probability distribution of precipitation intensity measurement at q by sensors at x_t. The expected value of Equation 6.2 with respect to $\boldsymbol{I}$ is equivalent to the rainfall intensity experienced at the location q. Because the effective range of the wipers is limited, we account for the probability of detection as a function of the distance between the sensor and the target. We denote by D_t the event that sensors detect the target, and by $\overline{D}_t$ the event that sensors fail to detect the target at time t. The probability of detecting a target located at q by sensors located at x_t, $p(D_t|q, x_t)$, is taken to decay with increasing distance to the sensor. Using the law of total probability, the conditional probability of a correct measurement is then given by:

$$p(M_t|\boldsymbol{z}, x_t) = p(M_t|\boldsymbol{z}, D_t, x_t)p(D_t|q, x_t) + p(M_t|\boldsymbol{z}, \overline{D}_t, x_t)p(\overline{D}_t|q, x_t) \tag{3.2}$$

where D_t is conditionally independent of $\boldsymbol{I}$ when conditioned on q. For example, consider $x_t = (0, 0)$, and $q = (q_1, q_2)$. If the decay function is taken to be a 2D Gaussian centered at x_t with covariance matrix $\sigma\mathbf{I}$ where $\mathbf{I}$ is a 2 by 2 identity matrix, then:

$$p(D_t|q, x_t) = \widetilde{\eta}_t \frac{1}{2\pi\sigma^2} \exp\left(-\frac{q_1^2 + q_2^2}{2\sigma^2}\right) \tag{3.3}$$

Where $\widetilde{\eta}_t$ is a normalization constraint. If the target is *not* detected (i.e., $\overline{D}_t$), then the measurement is assumed to be unreliable, and the likelihood, $p(M_t|z, \overline{D}_t, x_t)$, is modeled using a prior distribution. If there is no prior information available, the function is modeled using a uniform distribution. Now let $b_t(z)$ represent the posterior probability of the precipitation intensity given a target location q at time t. Using Bayes' Theorem, $b_t(z)$ can be formulated:

$$b_t(z) = \eta_t p(M_t|z, x_t) b_{t-1}(z), \qquad t = 1, 2, \ldots \tag{3.4}$$

Where η_t is a normalization constant and b_0 is uniform if no information is available at $t = 0$. This filtering equation forms the basis of the rainfall field updating algorithm. To reduce computational complexity, the filtering operation is implemented using a Sequential Importance Resampling (SIR) Particle Filter [111].

The results of the Bayesian sensor fusion procedure are evaluated by determining the proportion of instances where the combined data product is able to predict the binary rainfall state. We characterize the true and false positive rates for the largest storm event (2014-08-11) using an iterated "leave-one-out" cross-validation approach. First, a single vehicle is removed from the set of vehicles. The Bayesian update procedure is then executed using all vehicles except the excluded vehicle, and an updated rainfall map is generated. Next, the rainfall states predicted by the corrected rainfall field (radar and wiper) and the original rainfall field (radar only) are compared against the rainfall states predicted by the omitted vehicle. The performance of each data product is evaluated based on its ability to reproduce the binary rainfall state observed by the omitted vehicle. Performing this process iteratively yields the true and false positive rates for both the original (radar only) and updated (radar and wiper) rainfall fields. This procedure is repeated for each vehicle in the set of vehicles to generate Receiver-Operator Characteristic (ROC) curves, which characterize the true and false positive rates across an ensemble of simulations.

Published version

This chapter was reprinted with permission from the article [112]:
Bartos M, Park H, Zhou T, Kerkez B, Vasudevan R. (2019) *Sci. Rep.*, **9(1)**, 170

Acknowledgements

Funding for this project was provided by MCubed (grant 985), the Ford Motor Company–University of Michigan Alliance (grant N022977), and the University of Michigan. Vehicle metadata and camera footage are provided courtesy of the University of Michigan Transportation Research Institute (UMTRI). We would like to thank UMTRI Director Jim Sayer and UMTRI Lead Engineer Scott Bogard for helping to obtain the vehicle sensor data used in this study.

Author contributions statement

Matt Bartos wrote the paper, performed the analysis, and helped with the implementation of the filtering algorithm. Hyongju Park developed, implemented, and validated the filtering algorithm. Tian Zhou analyzed the dashboard camera data and assisted with analysis of the windshield wiper data. Branko Kerkez and Ram Vasudevan originated the concept of the study, guided the development of the methods, and assisted in writing the paper. Additional inspection and labeling of vehicle dashboard footage was performed by Aditya Prakash Singh. All authors reviewed the manuscript.

Additional information

Code and data for this study are available at: `github.com/kLabUM/vehicles-as-sensors`.

Supplementary Information

S3.1 Supplementary note on binary detection performance

Binary detection performance is sensitive to a number of factors, including the temporal resolution of the ground truth data and the configuration of wiper sensors. While these factors can affect the magnitude of binary classification performance, under all scenarios considered, wiper measurements are a better detector of the binary rainfall state than either radar or gage measurements.

Binary detection performance can be affected by the temporal resolution at which the ground truth data is collected. To ensure robustness, labeling of vehicle footage was performed independently by two reviewers. The first reviewer labeled the observed rainfall state for each vehicle over all three days of the study period (2014-06-12, 2014-06-28, 2014-08-11) at a temporal interval of 1 minute. A second reviewer labeled the observed rainfall and wiper state for the largest storm event (2014-08-11) at an enhanced time resolution of roughly 3 seconds. Due to the time-intensive nature of labeling video data at this temporal resolution, and due to the strong agreement between the two labeled datasets, this second round of labeling was not performed for the remaining two days (2014-06-12 and 2014-06-28). Despite the difference in time resolution, manual labeling of the video data showed strong agreement. Taking the high-resolution dataset to represent the ground truth rainfall state (and aggregating the high-resolution dataset to the temporal resolution of the low-resolution dataset), the true positive rate of the low temporal-resolution camera observations was 92.6%, while the true negative rate was 99.3%. Agreement in terms of positive detection was lower due to the difference in temporal resolution between the two sources. The low-resolution camera observation dataset classifies each minute-long interval as either “raining” or “not raining”. However, the high-resolution ground truth dataset contains many instances in which part of a given minute-long interval contains rain, and part does not. Thus, when the high-resolution dataset is aggregated to match the resolution of the low-resolution dataset, there are more intervals where some amount of rain is detected (yielding more instances of positive detection overall). A similar mismatch occurs if the low-resolution dataset is interpolated to match the time resolution of the high-resolution dataset. This time resolution mismatch also affects comparisons between the ground truth and other data sources (e.g. wiper, radar and gages). In general, the difference in classification performance between data sources decreases when the ground truth dataset is aggregated in time. Differences in classification performance become more pronounced when a high-resolution ground truth dataset is used.

Many vehicles exhibited data quality issues such as non-reporting wiper sensors, malfunctioning wiper sensors, or unobservable wiper modes. These data quality issues may impact the performance of the wiper as a classifier, but are largely attributable to the fact that the data is taken from a pilot study in which sensor configurations are not standardized. For some vehicles, wiper sensors were simply not configured to report wiper data. In these instances, the reported wiper value was zero for the entire observation period even though wiper movement was observed during manual inspection of the dashboard footage. Vehicles for which wiper sensors were not configured were removed from the analysis. Other vehicles exhibited malfunctioning or poorly configured sensors. For instance, in some cases the wiper intensity fluctuated between 0 and 1 at a frequency on the order of milliseconds—a behavior which is clearly not possible for a human driver. Video footage confirmed that the sensor was malfunctioning during these time periods. Malfunctioning vehicles were also removed from the analysis. Perhaps the most common data quality issue, however, is that several vehicles exhibited unobservable wiper modes. In this case, sensors were configured to report some wiper intensity states but not others. For example, the sensor may report the wiper intensity when the wiper switch is in a "continuous" mode, but may not report the wiper intensity when the wiper is placed in a manual "wipe" mode. These cases could only be detected by manual inspection of the camera footage. These data issues can largely be attributed to the fact that the sensor data is taken from a pilot study in which sensor configurations vary from vehicle to vehicle. As manufacturers standardize sensor configurations for connected vehicles, the relevance of these issues is likely to diminish.

The performance of the wiper as a classifier can be improved by (i) comparing wiper data against a ground truth dataset obtained at a high temporal resolution, and (ii) correcting errors in the wiper sensor readings. When manual observations of the wiper state are used to correct unobservable wiper modes, and the resulting corrected wiper data is compared to the 3-second resolution camera observations, the binary classification performance over weather radar is significantly enhanced: the true positive rate of the wiper data is 5.2% higher than radar, while the true negative rate is 7.7% higher. Table S3.1 shows the true and false positive rates for all technologies (during the 2014-08-11 storm event) when these two conditions are met.

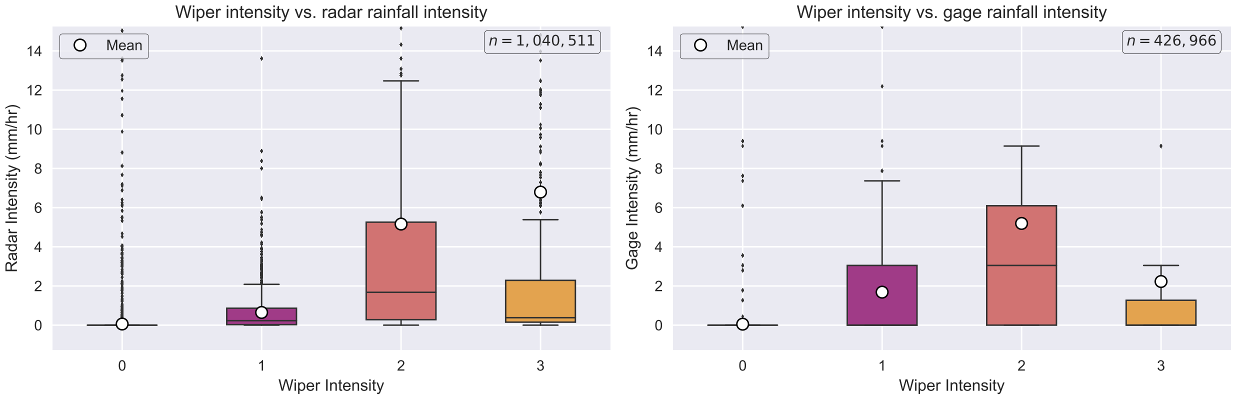


Figure S3.1. Comparison of radar, gage and wiper intensities. Data correspond to three storm events on 6/12/2014, 6/28/2014, and 8/11/2014. The left panel shows the distribution of radar precipitation measurements associated with each wiper intensity. The right panel shows the distribution of gage precipitation measurements associated with each wiper intensity for vehicles located within 2 kilometers of the gage. Note that the range limitation reduces the number of data points available. No clear relationship is observed between wiper intensity and rainfall intensity.

Table S3.1. Classification performance with high-resolution ground truth data. These binary performance metrics hold when (i) ground truth observations at a resolution of 2.4 seconds are used, and (ii) manual corrections are made to the wiper state according to the wiper state in the observed camera footage (i.e. unobservable wiper modes are corrected).

Metric	Gage	Radar	Wiper
True Positive Rate (%)	55.1	91.8	97.0
True Negative Rate (%)	96.9	87.4	95.1

Chapter 4

Pipedream: An Interactive Digital Twin Model for Urban Drainage Systems

Abstract

Faced with growing water infrastructure challenges, many cities are now seeking to build *digital twins* of urban stormwater systems that combine sensor data with online models in order to better understand and control system dynamics. Towards this goal, this study presents `pipedream`—an end-to-end software toolkit for real-time modeling and state estimation in urban stormwater networks. The toolkit combines (i) a new hydrodynamic solver based on the full one-dimensional Saint-Venant equations and (ii) an implicit Kalman filtering methodology that efficiently updates system states based on observed data. Drawing on sensor data from a real-world stormwater network, we find that the state estimation toolkit is effective at both interpolating system states and forecasting future states based on current measurements. By providing a complete, real-time view of stormwater system dynamics, this toolkit will enable better evaluation of system performance, improved detection of hazards, and more robust implementation of real-time control.

4.1 Introduction

In the wake of growing urban populations, aging infrastructure, and more frequent extreme weather events, many cities are struggling to manage stormwater-related challenges such as flash flooding and combined sewer overflows [19]. Engineers have traditionally responded to these challenges by expanding stormwater infrastructure and implementing best management practices [113]. However, despite the high costs of these interventions, it is often difficult to evaluate their benefits and impacts due to (i) a lack of real-time data and (ii) a limited understanding of system dynamics [19, 87]. Interventions may fail to achieve performance targets due to design oversights, insufficient maintenance, or changing hydrologic conditions [18, 114]. In certain pathological cases, measures aimed at improving flood control and urban water quality may actually worsen the problems they are intended to solve [57, 115]. In the absence of continuous monitoring, these issues may remain undetected until negative impacts have already accrued.

To ensure that stormwater systems achieve desired outcomes, water managers are now seeking to build *digital twins* of real-world networks that use embedded sensors and online models to monitor system dynamics in real-time. While these digital twins vary widely in scope and sophistication, they share the general goal of integrating sensor data, hydrodynamic models, and data assimilation techniques in order to assess system performance and determine effective control strategies. Using these technologies, operators can detect abnormal conditions within sewer networks and then dispatch maintenance crews to make repairs before damage occurs [12, 18]. Moreover, when combined with real-time control (RTC), continuous monitoring has shown impressive results in reducing combined sewer overflows [12, 20–23], reducing operational costs [12, 18], and improving urban water quality [24–26, 101].

While digital twins promise to address many of stormwater management's biggest pain points, adoption has been hampered by a lack of sufficient tools and theory for online hydraulic modeling and real-time data assimilation. In practice, hydraulic models are primarily executed in batch mode for the purposes of sizing pipes and evaluating infrastructure expansions [39]. Popular stormwater models are oriented towards this use case, and relatively few software packages provide support for online modeling or data assimilation. Consequently, for real-world systems that seek to implement continuous monitoring, the underlying process model is often substantially simplified—for instance, assuming that the stormwater network behaves as a series of cascading linear reservoirs [22]. These simplifications may introduce uncertainty when interpolating or forecasting system states, which in turn may impair the performance of real-time control strategies. New tools are needed to enable online state estimation and control capabilities while at the same time ensuring that system dynamics are accurately represented.

To pave the way for continuous monitoring and control of urban drainage systems, we introduce `pipedream`—a software toolkit for building digital twins of stormwater networks. This toolkit consists of (i) an efficient hydraulic solver based on the full one-dimensional Saint-Venant equations and (ii) an implicit Kalman filtering approach that updates system states based on streaming sensor data (see Figure 4.1). Using the full one-dimensional Saint-Venant equations ensures that the model adequately captures the physics of a wide variety of real-world stormwater networks. Similarly, Kalman filtering enables fast assimilation of sensor data while at the same time respecting the dynamics of the physical system. `pipedream` can be run in online mode, advancing forward in sync with the real-world system and assimilating sensor measurements in real-time. Moreover, the toolkit provides a robust interface for executing dynamic controls through the use of adjustable orifices, weirs and pumps. Taken together, the software described in this paper provides an end-to-end framework for real-time monitoring and control that will enable system operators to better manage stormwater infrastructure.

4.2 Prior Work

Previous efforts towards interactive stormwater modeling have largely focused on developing enhancements to existing models, such as the EPA Stormwater Management Model (SWMM) [116]. One of the earliest of these efforts was MATSWMM: a MatLab/Python library aimed at simulating real-time control strategies for stormwater systems [88]. The MATSWMM library provides bindings for the SWMM hydrodynamic solver, and also offers tools for implementing control strategies; however, at the time of writing, the project is no longer actively maintained. pySWMM is an actively-maintained Python library that also provides interactive bindings for SWMM, allowing users to modify sys-

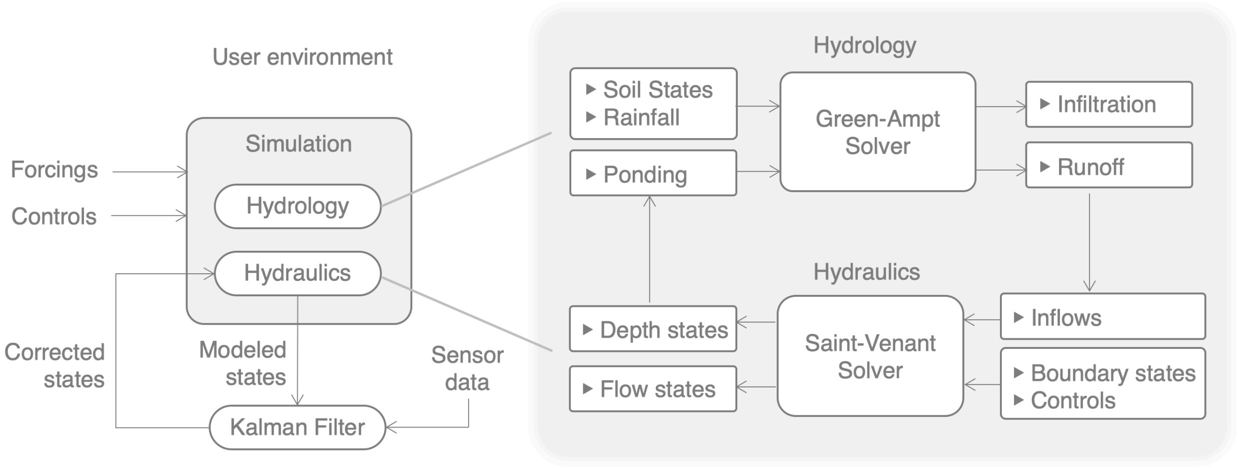


Figure 4.1. Software diagram of the `pipedream` toolkit. The simulation engine consists of a coupled hydraulic/hydrologic solver based on the Saint-Venant and Green-Ampt equations. The solver simulates stormwater dynamics in real-time by combining rainfall forcings, user-specified control inputs, and online sensor data which is fused into the model using a Kalman Filter.

tem states and simulate real-time control strategies [89]. While pySWMM has been used to great effect in simulating real-time control strategies within industry, its application to continuous monitoring has been limited by the fact that SWMM does not offer a state-space representation of system dynamics, making it difficult to apply model-based data assimilation techniques like Kalman Filtering.[4]

On the data assimilation side, a significant body of work has explored the problem of state/parameter estimation in hydrologic systems. Much of this research focuses on the problem of streamflow forecasting in large river basins [119]. These studies seek to improve streamflow forecasts by using data assimilation techniques to correct initial soil moisture conditions [120–122], snow water storage [123], and rainfall forcings [120]. A variety of data assimilation approaches have been investigated, including variational data assimilation [120], best linear unbiased estimation [121], particle filtering [123, 124], and ensemble Kalman filtering [122, 123, 125]. While most of the literature focuses on offline data assimilation, a few studies have examined data assimilation in a real-time context. Schwanenberg (2011) develop a data assimilation approach based on Ensemble Kalman filtering to assist with real-time control of large delta river systems [126]. However, their approach uses the kinematic wave approximation of the Saint-Venant equations, rendering it unsuitable for stormwater networks with bidirectional flow.

[4] Although it is possible to assimilate data using a surrogate model when the underlying process model is unavailable [117], this approach often lacks the performance guarantees of model-based data assimilation approaches [118].

In the context of stormwater networks, the application of data assimilation has been more limited. In practice, efforts towards incorporating sensor data into stormwater models have mainly focused on model parameter calibration [127–131]. Online state estimation has generally been achieved through model-specific deterministic methods that are difficult to reconcile with modern state estimation theory [132, 133]. Studies that explore more formal state estimation techniques generally rely on simplified conceptual models. Hutton et al. (2014), for instance, present a method for applying a Kalman Filter to a simplified stormwater model based on a series of cascading linear reservoirs [134], while Breinholt et al. (2011, 2012) investigate the application of Extended Kalman Filtering to a lumped conceptual urban rainfall-runoff model [135, 136]. While this line of research shows the potential for data assimilation to improve our understanding of stormwater system performance, more work is needed to integrate robust data assimilation techniques with our best knowledge of system dynamics, as represented by physically-based models. To our knowledge, there is currently no fully physically-based interactive stormwater model that provides real-time data assimilation capabilities. To fill this need, we present `pipedream`—a new digital twin model for stormwater networks.

4.3 Methods

In this section, we describe the development of the `pipedream` toolkit, including the theory behind the hydrodynamic model, its numerical implementation, and the development of a state estimation procedure for fusing real-time sensor data. First, we develop and implement a robust hydraulic solver that facilitates data assimilation by providing a state space model of the stormwater system. Next, we combine the hydraulic solver with an implicit Kalman filter to facilitate real-time assimilation of sensor data into the dynamical model. Finally, drawing on sensor data from a real-world stormwater network, we test the data assimilation framework by evaluating its efficacy at interpolating and forecasting system states.

Overview of the hydraulic solver

To enable real-time state estimation in urban drainage systems, we first develop and implement a new hydraulic solver that (i) enables real-time interactive usage, and (ii) provides a state space representation of system dynamics. The hydraulic solver developed in this study is based on an implicit staggered-grid scheme for solving the one-dimensional Saint-Venant equations in sewer/channel networks [40]. This formulation has demonstrated impressive accuracy and stability in previous applications, and variants of the scheme are used in a number of proprietary software packages, including GSSHA and SEWERCAT [137, 138]. More crucially for this study, this numerical scheme facilitates the use of robust state estimation techniques like Kalman filtering by enabling the system dynamics to be cast in the form of an implicit state space model. In this section, we discuss the basic solver formulation, and show how the model can be reinterpreted as an implicit linear time-varying (LTV) state space system. We also present several improvements to the original solver formulation that enhance model stability, enable representation of backwater effects, and extend the applicability of the model to systems with dynamic control structures.

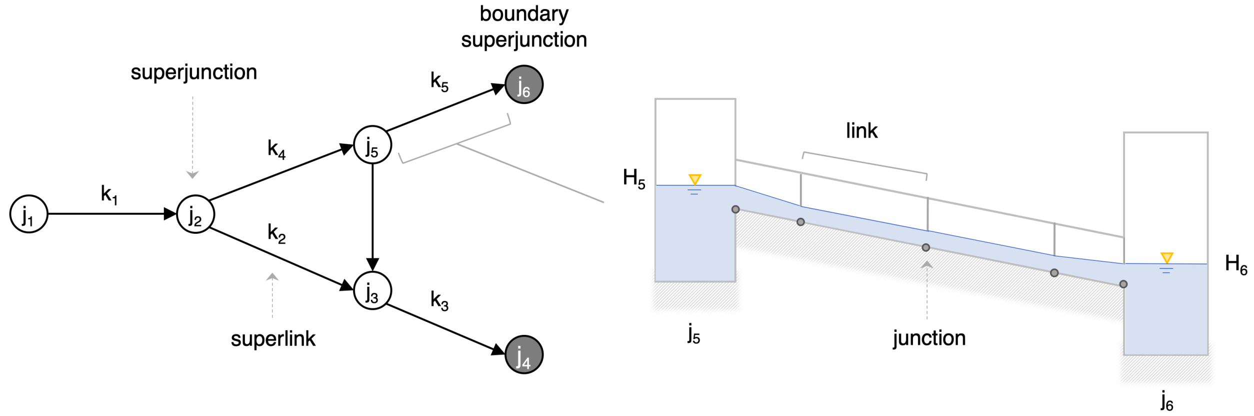


Figure 4.2. Hydraulic model structure. Example network from Ji (1998) demonstrating the fundamental computational elements used by the model [40], including superlinks/superjunctions (left), and links/junctions (right).

The hydraulic solver is a finite difference model with four distinct types of computational elements: links, junctions, superlinks, and superjunctions [40]. *Links* are finite volumes that represent sections of conduit or open channel. *Junctions* connect links together, and may represent manholes, grade changes, or simply extra computational elements. *Superlinks* are collections of links joined end-to-end by junctions with no branching. *Superjunctions* connect one or more superlinks together, and may represent storage basins, branching locations, or invert discontinuities. To facilitate the computation of pressurized flow, each link is equipped with a Priessman slot—a fictitious narrow groove located at the crown of the link that allows the one-dimensional unsteady open-channel flow equations to be applied to surcharged pipes [139]. Figure 4.2 shows an example network adapted from Ji (1998) [40], with all basic elements of the model labeled.

Flows within the channel network are modeled using the one-dimensional Saint-Venant equations. This pair of nonlinear partial differential equations consists of two parts: the continuity equation (4.1), which describes the mass balance for a finite volume, and the momentum equation (4.2), which describes the force balance:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_{in} \tag{4.1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(Qu) + gA\left(\frac{\partial h}{\partial x} - S_0 + S_f + S_L\right) = 0 \tag{4.2}$$

Where Q is discharge; A is the cross-sectional area of flow; u is the average velocity; h is depth; x is distance; t is time; q_{in} is the lateral inflow per unit width; and S_o, S_f and S_L represent the channel bottom slope, friction slope and local head loss slope, respectively. Using a staggered grid formulation, the continuity equation is applied to each junction (indexed by Ik), while the momentum equation applied to each link (indexed by ik). The equations are discretized using a Backward Euler-type implicit scheme (see Sections S4.2 and S4.3 in the Supplementary Information for derivations):

$$Q_{ik}^{t+\Delta t} - Q_{i-1k}^{t+\Delta t} + \left(\frac{B_{ik}\Delta x_{ik}}{2} + \frac{B_{i-1k}\Delta x_{i-1k}}{2} + A_{s,Ik}\right) \cdot \frac{h_{Ik}^{t+\Delta t} - h_{Ik}^{t}}{\Delta t} = Q_{in,Ik} \tag{4.3}$$

$$\begin{aligned} (Q_{ik}^{t+\Delta t} - Q_{ik}^{t})\frac{\Delta x_{ik}}{\Delta t} + u_{I+1k}Q_{I+1k}^{t+\Delta t} - u_{Ik}Q_{Ik}^{t+\Delta t} \\ +gA_{ik}(h_{I+1k}^{t+\Delta t} - h_{Ik}^{t+\Delta t}) - gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(S_{f,ik} + S_{L,ik})\Delta x = 0 \end{aligned} \tag{4.4}$$

Where B is the top width of flow, A_s is the junction surface area, and Q_{in} is the exogenous flow input. The boundary conditions for each superlink are supplied by the upstream and downstream superjunction heads, assuming weir-like flow at the superlink inlet and outlet:

$$Q = CA\sqrt{2g\Delta H} \tag{4.5}$$

Where C is the inlet/outlet discharge coefficient, and ΔH is the difference in head between the superjunction and the adjacent superlink boundary junction.

The hydraulic model solves for all unknowns simultaneously at each time step by embedding the solutions to the Saint-Venant equations into a system of implicit linear equations wherein all unknowns are defined in terms of the unknown superjunction heads.

1. First, the discretized Saint-Venant equations are reformulated into recurrence relations that relate junction heads and link flows within each superlink.
2. The assumption of orifice-like flow between superjunctions and superlinks is used to establish boundary conditions for the superlink inlets and outlets.
3. Combining the recurrence relations together with the superlink boundary conditions, the system is reformulated as a sparse matrix equation with all unknowns expressed in terms of the unknown superjunction heads.
4. After solving for the unknown superjunction heads, the internal depths and flows within each superlink are recovered by substituting the superjunction boundary heads into the previously-developed recurrence relations.

This solution procedure affords a balance between stability and computational efficiency: while the implicit discretization scheme helps to ensure stability, the use of recurrence relations reduces the size of the solution matrix and helps to increase computation speed compared to the more conventional four-point implicit scheme [40]. However, in addition to its computational advantages, this scheme also facilitates data assimilation by enabling the solver to be cast in the form of a linear time-varying state space system.

Constructing the state space system

Through derivation of the solution matrix equation, we show that the hydraulic model is equivalent to a linear time-varying (LTV) state space system, which in turn allows for the application of powerful algorithms from signal processing and control theory, such as feedback control, modal analysis, and—as we show in this paper—Kalman filtering. In the following section, we derive the solution matrix equation and show how it is equivalent to a LTV state space system. Detailed derivations, including the discretization of governing equations and the development of the recurrence relations, are omitted from this section and can instead be located in the Supplementary Information (SI) document.

The solution matrix equation is derived by applying the continuity equation to all superjunctions, and then using recurrence relations to express all unknown variables in terms of the superjunction heads. Focusing on a single superjunction j, the continuity equation requires that the change in storage volume over a time interval Δt equal the sum of inflows minus outflows:

$$\sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} - \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t} + Q_{in,j}^{t+\Delta t} = \frac{A_{sj}(H_j^{t+\Delta t} - H_j^t)}{\Delta t} \tag{4.6}$$

Where Q_{dk} is the discharge at the downstream end of superlink k, Q_{uk} is the discharge at the upstream end of superlink k, $Q_{in,j}$ is the external inflow into superjunction j, A_{sj} is the surface area of superjunction j, and H_j is the total head at superjunction j. NKD_j represents the number of superlinks with their downstream ends attached to superjunction j, while NKU_j represents the number of superlinks with their upstream ends attached to superjunction j (thus, the first two terms represent the sum of inflows minus outflows from all superlinks attached to superjunction j).

Using the recurrence relations developed in SI Section S4.4, the discharge at the upstream and downstream ends of each superlink can be described using the following linear functions of the upstream and downstream superjunction heads at time $t + \Delta t$:

$$Q_{uk}^{t+\Delta t} = \alpha_{uk} H_{juk}^{t+\Delta t} + \beta_{uk} H_{jdk}^{t+\Delta t} + \chi_{uk} \tag{4.7}$$

$$Q_{dk}^{t+\Delta t} = \alpha_{dk} H_{juk}^{t+\Delta t} + \beta_{dk} H_{jdk}^{t+\Delta t} + \chi_{dk} \tag{4.8}$$

Where H_{juk} represents the head at the superjunction upstream of superlink k, and H_{jdk} represents the head at the superjunction downstream of superlink k. The α, β and χ coefficients are functions of the depths and flows inside each superlink that incorporate the solutions to the continuity and momentum equations in both the forward and backward directions. Substituting these linearized expressions into the continuity balance for superjunction j yields the linear equation:

$$\begin{aligned}\Big(\frac{A_{sj}}{\Delta t} + F^k_{j,j}\Big) \cdot H_j^{t+\Delta t} + \sum_{\ell=1}^{NKDj} \Phi^k_{j,juk_\ell} H^{t+\Delta t}_{juk_\ell} + \sum_{m=1}^{NKUj} \Psi^k_{j,jdk_m} H^{t+\Delta t}_{jdk_m} \\ = \frac{A_{sj}}{\Delta t} H_j^t + Q^{t+\Delta t}_{in,j} + G^k_j\end{aligned} \tag{4.9}$$

Where:

$$F^k_{j,j} = \sum_{m=1}^{NKUj} \alpha_{uk_m} - \sum_{\ell=1}^{NKDj} \beta_{dk_\ell} \tag{4.10}$$

$$\Phi^k_{j,juk_\ell} = -\alpha_{dk_\ell} \tag{4.11}$$

$$\Psi^k_{j,jdk_m} = \beta_{uk_m} \tag{4.12}$$

$$G^k_j = \sum_{\ell=1}^{NKDj} \chi_{dk_\ell} - \sum_{m=1}^{NKUj} \chi_{uk_m} \tag{4.13}$$

Applying these equations to each superjunction j yields the following implicit state space equation:

$$\big(A_K(t) + A_S(t)\big) \cdot \mathbf{x}(t + \Delta t) = A_S(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + D_K(t) \tag{4.14}$$

Where $\mathbf{x}(t)$ is the state vector of superjunction heads at the current time step and $\mathbf{x}(t+\Delta t)$ is the state vector of superjunction heads at the next time step. The elements of $A_K(t)$ are defined such that $A_{K_{j,j}}(t) = F^k_{j,j}$; $A_{K_{j,juk_\ell}} = \Phi^k_{j,juk_\ell}$; and $A_{K_{j,jdk_m}} = \Psi^k_{j,jdk_m}$ (with all other elements of $A_K(t)$ equal to zero). Similarly, the elements of $D_K(t)$ are defined such that $D_{K_j} = G^k_j$. $A_S(t)$ is a diagonal matrix representing the current superjunction storage:

$$A_S(t) = \text{diag}\bigg(\frac{A_{sj}}{\Delta t}\bigg), \quad \forall j = 1, 2, ..., M \tag{4.15}$$

The exogenous input $B(t)\mathbf{u}(t)$ represents the external inflow (e.g. runoff) to each superjunction j. Assuming that each superjunction has its own independent input signal, the input transition matrix is the identity matrix:

$$B(t) = I \tag{4.16}$$

And the input signal is equal to the external inflow at each superjunction:

$$\mathbf{u}(t) = [Q_{in,1},\ Q_{in,2}, \cdots Q_{in,M}]^T \tag{4.17}$$

To illustrate the construction of the solution matrix in concrete terms, we include here the solution matrix for the example network in Figure 4.2. For brevity, define $\tilde{F}_{j,j} = \frac{A_{sj}}{\Delta t} + F^k_{j,j}$ and $\tilde{G}_j = \frac{A_{sj}}{\Delta t} H^t_j + Q_{in,j} + G^k_j$. Thus, for the example network in Ji (1998) [40], the sparse matrix equation at time $t + \Delta t$ is expressed as:

$$\begin{bmatrix} \tilde{F}_{1,1} & \Psi^k_{1,2} & 0 & 0 & 0 & 0 \\ \Phi^k_{2,1} & \tilde{F}_{2,2} & \Psi^k_{2,3} & 0 & \Psi^k_{2,5} & 0 \\ 0 & \Phi^k_{3,2} & \tilde{F}_{3,3} & \Psi^k_{3,4} & \Phi^k_{3,5} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \Phi^k_{5,2} & \Psi^k_{5,3} & 0 & \tilde{F}_{5,5} & \Psi^k_{5,6} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H^{t+\Delta t}_1 \\ H^{t+\Delta t}_2 \\ H^{t+\Delta t}_3 \\ H^{t+\Delta t}_4 \\ H^{t+\Delta t}_5 \\ H^{t+\Delta t}_6 \end{bmatrix} = \begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \\ H^{t+\Delta t}_{bc,4} \\ \tilde{G}_5 \\ H^{t+\Delta t}_{bc,6} \end{bmatrix} \tag{4.18}$$

This sparse system is solved for the unknown left-hand vector of superjunction heads at time step $t+\Delta t$. Once the superjunction heads are known, the depths and discharges at the upstream and downstream superlink boundaries are computed. Finally, the recurrence relations are used to solve for the internal depths and flows within each superlink.

Implicit Kalman Filter

Having defined the system dynamics in terms of an implicit LTV system, an implicit Kalman Filter is developed to fuse sensor data with the dynamical model. Kalman Filtering is a recursive Bayesian estimation algorithm that (i) uses a dynamical system model to generate a prior estimate of system states, then (ii) updates this prior with observed data to produce a posterior estimate [140]. It can be shown that the Kalman Filter is the optimal linear estimator for system states when the dynamical system model is perfectly accurate, and the noise is white and Gaussian with known covariance [140]. In the following section, we derive the Kalman recursion for an implicit LTV system, following the treatment provided in Skliar and Ramirez (1997) [141]. As a starting point, we may first express the internal states and observed outputs of the system in terms of a "hidden" state equation along with an observed output equation.

$$A_1(t)\mathbf{x}(t+\Delta t) = A_2(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + C(t)\mathbf{w}(t) + D(t) \tag{4.19}$$

$$\mathbf{z}(t+\Delta t) = H(t+\Delta t)\mathbf{x}(t+\Delta t) + \mathbf{v}(t+\Delta t) \tag{4.20}$$

Where $\mathbf{x}(t)$ is the $n-$dimensional state vector at the current time step, $\mathbf{u}(t)$ is an $\ell-$dimensional input vector, $\mathbf{w}(t)$ is a $p-$dimensional stochastic disturbance, $\mathbf{z}(t+\Delta t)$ is the $m-$dimensional observed state at the next time step, and $\mathbf{v}(t+\Delta t)$ is the m-dimensional vector of measurement noise. $A_1(t)$ and $A_2(t)$ are $(n \times n)$ state transition matrices, $B(t)$ is an $(n \times \ell)$ input transition matrix, $C(t)$ is an $(n \times p)$ stochastic disturbance transition matrix, $D(t)$ is an $(n \times 1)$ vector of constants, and $H(t+\Delta t)$ is the $(m \times n)$ measurement transition matrix.

The stochastic disturbance $\mathbf{w}(t)$, and measurement noise $\mathbf{v}(t+\Delta t)$ are assumed to be random vectors of zero-mean Gaussian white noise:

$$E[\mathbf{w}(t_2)\mathbf{w}(t_1)^T] = Q(t)\delta(t_2 - t_1) \tag{4.21}$$

$$E[\mathbf{v}(t_2+\Delta t)\mathbf{v}(t_1+\Delta t)^T] = R(t+\Delta t)\delta(t_2 - t_1) \tag{4.22}$$

Where $Q(t)$ is the $(p \times p)$ covariance matrix for the process noise and $R(t+\Delta t)$ is the $(m \times m)$ covariance matrix for the measurement noise. To apply a Kalman filter to a system of this form, we may first define the auxiliary state vector [141]:

$$\mathbf{y}(t+\Delta t) = A_1(t)\mathbf{x}(t+\Delta t) \tag{4.23}$$

An estimate of the auxiliary state vector at the next time step is produced by propagating the dynamical model forward in time:

$$\hat{\mathbf{y}}_{t+\Delta t|t} = A_2(t)\hat{\mathbf{x}}_{t|t} + B(t)\mathbf{u}(t) + D(t) \tag{4.24}$$

Next, the estimation error covariance of $\mathbf{y}(t + \Delta t)$ is computed by propagating the previous error covariance forward in time through the dynamical model:

$$P^{y}_{t+\Delta t|t} = A_2(t)P^{x}_{t|t}A_2^{T}(t) + C(t)Q(t)C^{T}(t) \tag{4.25}$$

The auxiliary measurement matrix H_1 is defined as:

$$H_1(t + \Delta t) = H(t + \Delta t)A_1^{-1}(t) \tag{4.26}$$

The optimal Kalman gain can then be computed:

$$L^{y}(t + \Delta t) = P_{t+\Delta t|t}H_1(t + \Delta t)\Big[H_1(t + \Delta t)P^{y}_{t+\Delta t|t}H_1^{T}(t + \Delta t) + R(t + \Delta t)\Big]^{-1} \tag{4.27}$$

The updated estimate of the auxiliary state vector is obtained by applying the Kalman gain to the forward propagation of the dynamical system model:

$$\hat{\mathbf{y}}_{t+\Delta t|t+\Delta t} = A_2(t)\hat{\mathbf{x}}_{t|t} + B(t)\mathbf{u}(t) + D(t) + L^{y}(t+\Delta t)\Big[\mathbf{z}(t+\Delta t) - H_1(t+\Delta t)\hat{\mathbf{y}}_{t+\Delta t|t}\Big] \tag{4.28}$$

The updated estimate of the original state vector is recovered through inversion of the left state transition matrix:

$$\hat{\mathbf{x}}_{t+\Delta t|t+\Delta t} = A_1^{-1}(t)\hat{\mathbf{y}}_{t+\Delta t|t+\Delta t} \tag{4.29}$$

The covariance measurement update for the auxiliary state vector is then computed as:

$$P^{y}_{t+\Delta t|t+\Delta t} = (I - L^{y}(t + \Delta t)H_1(t + \Delta t))P^{y}_{t+\Delta t|t} \tag{4.30}$$

Finally, the covariance measurement update for the original state vector is recovered through inversion of the left state transition matrix:

$$P^{x}_{t+\Delta t|t+\Delta t} = A_1^{-1}(t)P^{y}_{t+\Delta t|t+\Delta t}(A_1^{-1}(t))^{T} \tag{4.31}$$

After computing the updated covariance measurement, the process is repeated for the next time step, starting with equation 4.24 and ending with equation 4.31.[5]

The data fusion procedure proceeds as follows: at each time step, the hydraulic solver is advanced to determine the hydraulic heads and update the state transition matrices. Next, the Kalman recursion is applied to correct the hydraulic heads at time step $t + \Delta t$ based on observed data. Finally, the superlink boundary conditions and internal states are computed based on the updated superjunction heads. This process is repeated indefinitely (potentially in real-time) until the simulation is terminated.

Enhancements to the hydraulic solver

To enable representation of a broader array of real-world stormwater networks, we make substantial additions to the original numerical scheme. These changes allow for modeling of real-world stormwater networks that would be either impractical or impossible under the original formulation.

Control structures: To facilitate simulation and execution of real-time controls, we modify the numerical scheme to enable representation of orifices, weirs, and pumps (see Sections S4.9, S4.10, and S4.11). The governing equations for these control structures are embedded directly into the system solution matrix, facilitating the use of classical state estimation and control algorithms for LTV systems.

Mobile computational elements: The original numerical scheme suffers from instability when modeling backwater effects. We correct this problem by introducing mobile computational elements that track discontinuities in the water surface profile (see Section S4.12).

Accuracy improvements: High-gradient conditions were found to induce mass balance errors under the original formulation. To reduce these errors, we re-derive the recurrence relations and superlink boundary conditions to eliminate some error-generating assumptions (see Sections S4.4 and S4.5).

[5] Note that for sufficiently large state-space systems, A_1 should not be inverted directly [141]. Rather, equations 4.26, 4.29 and 4.31 should be solved implicitly for their dependent variables using algorithms for solving linear systems of equations (e.g. Gaussian Elimination).

Adaptive step size control: Implicit solvers typically maintain better stability at large time steps when compared to explicit solvers—especially during near-steady-state conditions. To exploit this strength, we implement an adaptive time-stepping algorithm that allows the solver to use a small timestep during highly transient conditions and a large timestep during weakly transient conditions (see Section S4.13) [142].

Infiltration/runoff coupling: To enable the use of `pipedream` as a standalone stormwater management model, we implement an infiltration module based on the well-known Green-Ampt formulation [143]. This module is coupled bi-directionally with the hydraulic solver to enable more accurate representation of overland flow and runoff generation (see Section S4.14).

Implementation

The `pipedream` toolkit is implemented in the Python programming language, which provides a powerful interpreter environment for interactive use [144]. Acceleration of numerical code is realized using the *numba* just-in-time compiler, which compiles native Python code to machine code that achieves speeds comparable to code written in C or FORTRAN [145]. We accelerate the solution of the system matrix equation by using a banded matrix solver: upon model initialization, the system matrix is automatically permuted into a banded form using the Reverse Cuthill-McKee algorithm [146]. All code and data for this study are available at `github.com/mdbartos/pipedream`.

Model validation

We assess the state estimation toolkit by applying our methodology to a real-world stormwater network and then evaluating the extent to which the Kalman filter improves the accuracy of interpolated and forecasted system states. First, a real-world catchment is selected, and real-time depth data is collected at four sites. We then construct a `pipedream` model of the catchment and force the model with a real-world storm event. We then use a holdout cross-validation approach to measure the extent to which fusing sensor data at selected sites reduces error at the holdout sites. We also evaluate the ability of the Kalman filter to forecast system states by fusing sensor measurements at one-hour intervals and quantifying the reduction in error over the remainder of each hour.

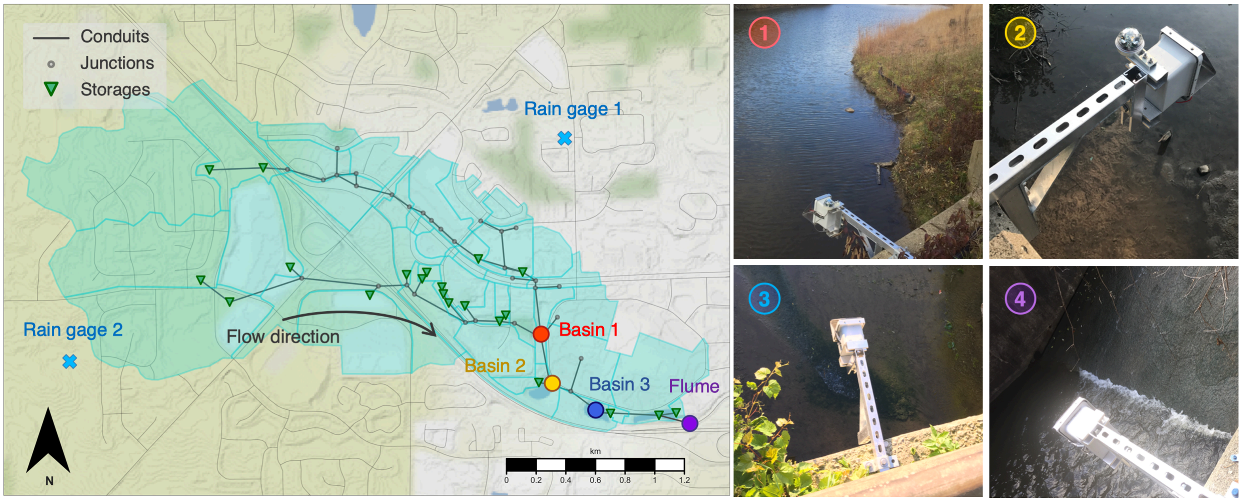


Figure 4.3. Overview of study area. Left: Map of stormwater network including the contributing area, channel network, hydraulic storage elements, and four sensor sites. Right: Photographs of sensor sites located on basins 1-3 and outlet flume.

These two applications—interpolation, and forecasting—represent important use-cases for real-time state estimation with continuous monitoring.

Our case study focuses on a 5.85 km^2 urban watershed located in the Midwestern United States (Fig. 4.3). This watershed is the subject of a long-term monitoring project led by the authors and thus features roughly two years of continuous sensor data [102]. Roughly 47% of the catchment is impervious, with a majority of the impervious area located towards the downstream half of the catchment. The representation of the channel network is derived from survey data and engineering drawings that describe the network topology, hydraulic geometries, storage curves, and various hydraulic and hydrologic parameters needed to properly model catchment dynamics.

To characterize the response of the catchment, wireless ultrasonic depth sensors are installed at four locations in series along the mainstem of the watershed (numbered in increasing order from 1 to 4 in the downstream direction). These sensors continuously report the distance to the water surface at an adaptive sampling rate that ranges from roughly 2 minutes to 1 hour, with a manufacturer-specified reading-to-reading error of approximately 1mm. Sensors at sites 1 and 2 measure the water level in two relatively large retention basins. The sensor at site 3 measures the water level in an outlet box directly downstream of a third retention basin. Overflow from this retention basin drains into the outlet box through a rectangular weir that is approximately 5m wide. Finally, the sensor at site 4 measures the water level at the downstream end of a rectangular flume. The rectangular flume is roughly 5m wide and is connected at the upstream end to a fourth retention basin. Raw sensor data is preprocessed using a combination of manual and automatic filtering techniques. Invalid readings and sensor spikes are flagged and removed using an automated filtering routine consisting of range checks and threshold tests on the second derivative of the sensor signal. After the initial quality control, distance measurements are converted to water depth estimates using field measurements of the sensor offset to the channel bottom.

We apply the new hydraulic model to a real-world storm event occurring on August 8, 2017. This rain event is selected because (i) all sensor sites were active and reporting during this storm, and (ii) the peak depth of the generated hydrograph is close to the median peak depth for the period of available data, meaning that the storm event is representative of a typical storm event in this location. Precipitation intensity data are collected from two weather stations operated by Weather Underground near the catchment of interest [107]. Runoff is generated from the rainfall using the Green-Ampt formulation and then fed into the hydraulic model. For the purposes of this analysis, we assume uniform rainfall intensity over the catchment and use the average intensity between the two gages as input to the model.

4.4 Results

Interpolation

Using a holdout cross-validation approach, we find that the Kalman filter is effective at interpolating system states at ungaged locations. Figure 4.4 shows the result of the holdout cross-validation assessment. For this experiment, the filter is applied to sensor sites 1 and 3, and the output of the updated model is compared with sensor measurements at sites 2 and 4. From Table 4.2, it can be seen that the filter reduces error at both holdout sites. While the model performs well on its own, the Kalman filter reduces the mean squared error (MSE) at site 2 by 25.5% and at site 4 by 17.9%. For site 2, a majority of the error is reduced at the peak of the hydrograph, while at site 4, a majority of the error is reduced at the falling limb. Because the filter improves model accuracy even

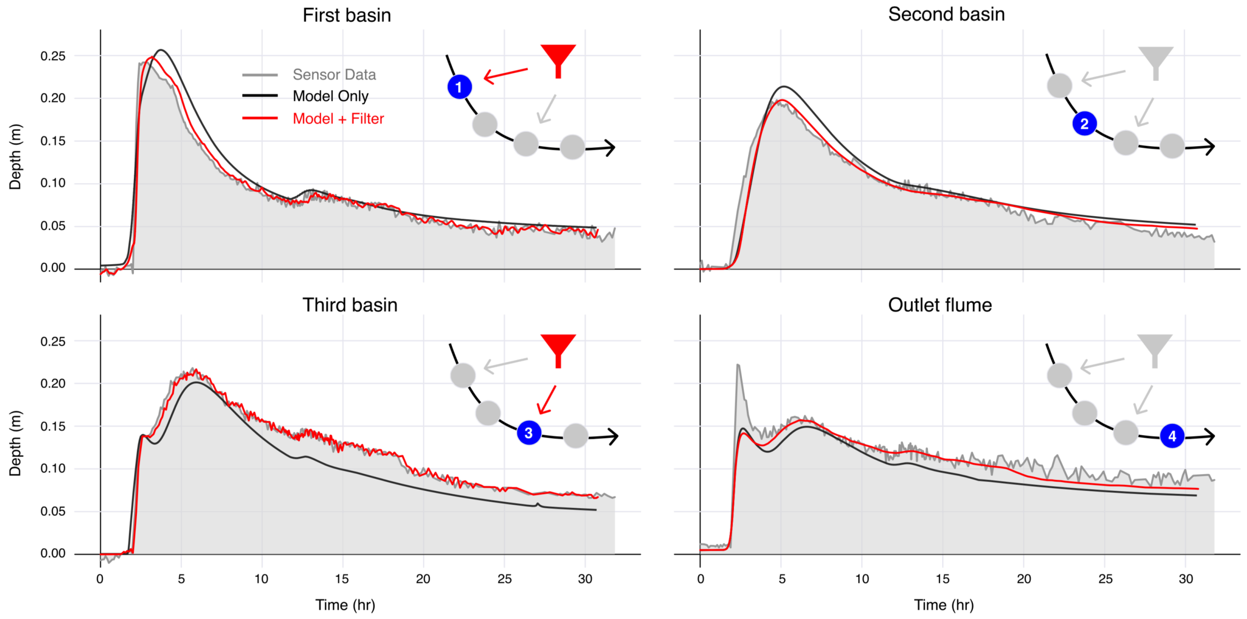


Figure 4.4. Validation of Kalman filter using holdout assessment. Left: Depth hydrographs at basins 1 and 3, where the Kalman filter is applied. Right: Depth hydrographs at holdout sites where Kalman filter was not applied. The Kalman filter reduces error at both holdout sites.

at locations where it is not directly applied, the holdout assessment suggests that the Kalman filter pushes the system closer to its actual state rather than simply "overfitting" individual sites to measured data.

The Kalman filter is capable of correcting error introduced by uncertain dynamical inputs, distinguishing it from a calibration-only approach. From Figure 4.4, it can be seen that the model without filtering over-predicts the discharge at sites 1 and 2, but under-predicts at sites 3 and 4. This result suggests that the spatial heterogeneity of runoff is a major source of error. This type of error is difficult to counter with model calibration alone, given that calibration tends to target static parameters of the system such as channel roughness coefficients and impervious area percentages. However, spatial variability in runoff may also result from spatial variability in the originating rainfall field. In contrast to continuous calibration, the Kalman filter handles this contingency by correcting system states in real-time, adding and removing mass from the system to match field observations. This approach is robust to both parameter and input uncertainty, making it especially suitable for real-time applications in which the driving input is often uncertain or unknown.

Table 4.2. Mean squared error of reconstruction at holdout sites.

	Basin 2	Outlet Flume
MSE (Model)	3.22E-4	6.36E-4
MSE (Filter)	2.40E-4	5.22E-4

Forecasting

By adaptively correcting system dynamics, Kalman filtering pushes the hydraulic system closer to its measured state and thus enables improved forecasting of system behavior. However, not all sites offer the same forecasting benefit, and sensor sites must be selected judiciously to maximize the accuracy of the forecast. Figure 4.5 shows the result of using the Kalman filter to forecast system states at 1-hour intervals. In this application, sensor data for each site is fused at the first minute of each hour (indicated by circular markers), and the model is then propagated forward in time to forecast system behavior for the remainder of the hour. The effectiveness of the forecast is then gaged based on the reduction in MSE, and the length of time that the correction persists.

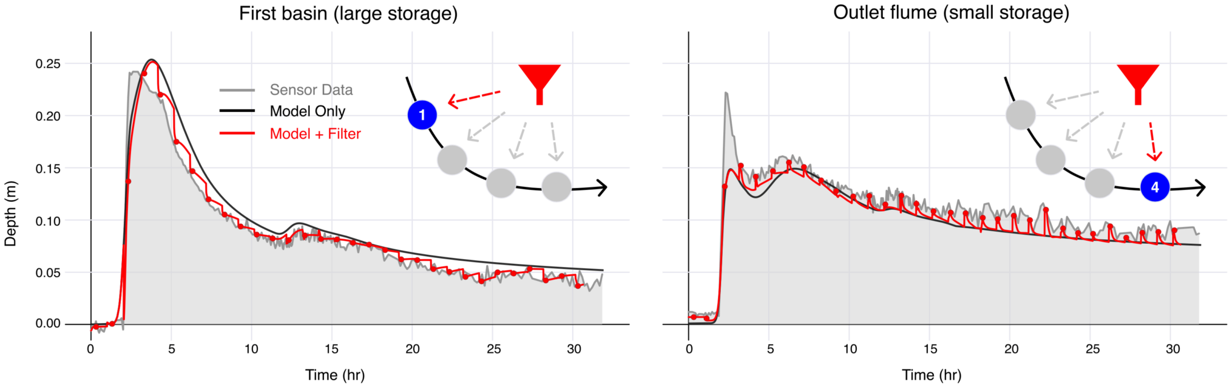


Figure 4.5. Forecasting hydraulic states at 1-hour intervals using the Kalman filter. Forecasting benefit is greater for sites with large storage capacity (left), and less so for sites with small storage capacity (right).

In general, the most effective sites for forecasting are those for which the volume of storage is large compared to the volume of water entering and exiting the control volume. When the filter is applied to correct the hydraulic head at a large retention pond (site 1), as shown in Figure 4.5 (left), the effect of the correction persists for the remainder of the hourly interval—in other words, applying the correction significantly changes the trajectory of the hydraulic head compared to the trajectory produced by the model alone. The correction imposed by the filter increases the accuracy of the forecasted states (measured by MSE) compared to the model-only forecast. Sites with larger storage capacity are most effective because small changes in hydraulic head result in relatively large changes to the mass and energy balance of the system, essentially allowing greater control over the state space.

By contrast, sites with smaller storage capacity offer less forecasting benefit, given that the effect of the correction is quickly overpowered by dynamics originating from elsewhere in the system. At the outlet flume (site 4), for example, the proportion of mass entering and leaving the control volume at each time step is relatively large compared to the mass within the control volume itself. Thus, correcting the system state at this location does not significantly alter the amount of mass or energy in the system, and the effect of the correction persists for only a short time after the filter is applied. As shown in Figure 4.5 (right), the trajectory of the hydraulic head quickly returns to the model-only trajectory after each application of the filter. Thus, sensor data from sites with little storage capacity must be fused at a rapid frequency (on the order of the time step used by the hydraulic solver) in order to produce noticeable forecasting benefit.

Computational Performance

The pipedream model offers significantly improved model run-times compared to pySWMM, an existing interactive solver for stormwater networks. Model run-time comparisons are conducted on networks α, β, and ϵ from the pystorms benchmarking library, with network sizes ranging from 26 to 210 nodes (superjunctions) [147]. When compared against pySWMM, pipedream ran between 15-190 times faster, with performance gains increasing for larger numbers of computational elements. For many networks, this multiple order-of-magnitude run-time improvement could mean the difference between offline usage and true real-time execution. When compared with SWMM5 in batch execution, pipedream is currently about 2.1-3.4 times slower for the benchmark scenarios chosen. However, pipedream was able to achieve more consistently stable results. In particular, for the β network, pipedream was able to achieve stable results, while SWMM was not—even when using a small timestep (0.1 s). Table 4.3 shows a detailed breakdown of model performance for all scenarios. Figures S4.4-S4.6 compare model outputs for pipedream and SWMM under each scenario.

Table 4.3. Performance of model runs under pipedream, SWMM and pySWMM. To assess the performance of pySWMM in an interactive context, pySWMM model runs incorporate the time needed to query depth and flow states from all nodes and links, respectively.

				Model run time (s)		
Scenario	Nodes	Links	Duration (hr)	pipedream	SWMM	pySWMM
α	26	30	12	4.42	1.29	64.9
β	210	210	24	42.4	19.98	8210
ϵ	78	77	960	196.0	64.8	10900

4.5 Discussion

By enabling real-time interpolation and forecasting of hydraulic states, `pipedream` provides a powerful new tool for urban flash flood nowcasting. The dynamics of urban flash flooding are complex and spatially heterogeneous, to the effect that there is no existing model that is capable of reliably forecasting flash floods in urban catchments [6]. While many cities use gage networks to help detect flooding and communicate flood alerts, gages are generally restricted to larger streams, leaving significant "blind spots" in the drainage network [148]. To address this problem, the data assimilation methodology presented in this paper will allow emergency managers to better estimate localized flooding at ungaged locations by interpolating hydraulic states from locations where sensor data is available. These high-resolution flood estimates will enable new forms of rapid flood response, such as localized alerts for motorists and targeted dispatch of emergency services. Moreover, by correcting system states in real-time, the `pipedream` toolkit will assist with flood forecasting, enabling emergency managers to more accurately predict downstream flooding at longer lead times—especially in cases where flooding is primarily driven by upstream transport.

In addition to detecting localized floods, the `pipedream` toolkit will also assist in identifying maintenance emergencies. Timely and accurate diagnosis of maintenance issues is essential for effective stormwater management. Short-term maintenance emergencies (such as storm drain blockages) can lead to localized flooding, while long-term maintenance issues (such as sediment accumulation) can degrade the overall performance of the stormwater system [18]. However, distinguishing true maintenance emergencies from spurious sensor faults remains a persistent challenge—especially when attempting to diagnose anomalies from sensor data alone [149]. `pipedream` handles this ambiguity by combining sensor data with a dynamical model of the stormwater system. By offering two independent estimates of system states, pipedream makes it easier to track the source of anomalies and differentiate true emergencies from false positives. Moreover, pipedream natively encodes the relative confidence in sensor/model outputs through the measurement/process covariances respectively, allowing users to incorporate prior information about sensor faults directly into the data assimilation procedure.

Perhaps most importantly, the pipedream toolkit provides a foundation for real-time control of urban drainage systems. Many cities are now implementing or seeking to implement real-time control systems for urban drainage systems in order to improve performance, cut costs, or mitigate stormwater-related hazards. In both modeling studies and real-world deployments, real-time control has shown proven results in mitigating combined sewer overflows and improving urban water quality [12, 20–26, 101]. However, effective real-time control is predicated on an accurate representation of system states. By fusing sensor data with an accurate physically-based process model, the pipedream toolkit provides a firm basis for control, whether for the purposes of simulation or real-world execution. Crucially, pipedream integrates control structures into the internal state-space model, facilitating native use of classical control algorithms such as model-predictive control and linear-quadratic regulation.

Conclusions

In this study, we develop a new toolkit for real-time *digital twinning* of urban stormwater systems. This toolkit consists of a robust hydraulic solver based on the full one-dimensional Saint-Venant equations along with an implicit Kalman filtering methodology that facilitates assimilation of real-time sensor data. Drawing on sensor data from a real-world stormwater network, we find that the implicit Kalman filter is effective at both interpolating system states within the network, and forecasting future states based on current measurements. By providing a physically-based methodology for state estimation in stormwater networks, this toolkit will enable system operators to pre-emptively detect and repair blockages, leaks and other maintenance emergencies. Moreover, by improving interpolation and forecasting of system states, our toolkit will provide a strong foundation for model-based real-time control schemes, such as model predictive control and linear quadratic regulation.

Acknowledgments

Support for this project was provided by the J. Robert Beyster Computational Innovation Graduate Fellowship, the National Science Foundation EarthCube initiative (Grant 1639640), and the National Science Foundation Smart and Connected Communities program (Grant 1737432).

Data availability

Code and data links are available at:

`https://github.com/mdbartos/pipedream`

Permanent DOI: `10.5281/zenodo.3950373`

Supplementary Information

S4.1 Glossary

Table S4.4. Glossary for hydraulic solver derivations.

Variable	Description
$a_{ik}, b_{ik}, c_{ik}, P_{ik}$	Momentum coefficients
a_h, q_q	Horizontal and vertical axis scaling factors for elliptical pump curve
A	Cross-sectional area of flow (general)
$A_1(t)$	Left-hand state transition matrix
$A_2(t)$	Right-hand state transition matrix
$A_K(t), A_S(t), A_O(t), A_P(t), A_W(t)$	Component state transition matrices
A_o	Maximum area of flow of orifice o
A_{cik}	Cross-sectional area of flow at control structure at link ik
A_{ik}	Cross-sectional area of flow at link ik
A_{dk}	Cross-sectional area of flow at downstream end of superlink k
$A_{s,Ik}$	Area of water surface at junction Ik
A_{sj}	Area of water surface at superjunction j
A_{uk}	Cross-sectional area of flow at upstream end of superlink k
$B(t)$	Input matrix
B_{ik}	Top width of flow at link ik
C_o	Coefficient of discharge of orifice o
C_{dk}	Coefficient of discharge at downstream end of superlink k
C_{ik}	Discharge coefficient for control structure in link ik
C_{uk}	Coefficient of discharge at upstream end of superlink k
C_{wR}	Rectangular weir discharge coefficient for weir w
C_{wT}	Triangular weir discharge coefficient for weir w
$D(t)$	Right-hand constants matrix
$D_K(t), D_O(t), D_W(t), D_P(t)$	Component right-hand constant matrices
D_{Ik}, E_{Ik}	Continuity coefficients
D^*_k	Intermediate coefficient
g	Acceleration due to gravity
h	Water depth (general)
h_{dk}	Water depth at downstream of superlink k
h_f	Cumulative infiltration depth
h_{Ik}	Water depth at junction Ik
h_{uk}	Water depth at upstream of superlink k
$H_{e,o}$	Effective orifice head
$H_{e,p}$	Effective pump head
$H_{e,w}$	Effective weir head
H_j	Head at junction j (invert elevation + water depth)
H_{jdk}	Head at junction downstream of superlink k
H_{juk}	Head at junction upstream of superlink k
H_{juo}	Head at junction upstream of orifice o
H_{jdo}	Head at junction downstream of orifice o
H_{jup}	Head at junction upstream of pump p
H_{jdp}	Head at junction downstream of pump p
H_{juw}	Head at junction upstream of weir w
H_{jdw}	Head at junction downstream of weir w

Table S4.4. Glossary for hydraulic solver derivations, continued.

Variable	Description
$H_{max,p}$	Maximum pump head
$H_{min,p}$	Minimum pump head
ΔH_{dk}	Head difference at downstream end of superlink k
ΔH_{uk}	Head difference at upstream end of superlink k
i_f	Infiltration rate
K_s	Saturated hydraulic conductivity
L_w	Length of transverse weir w
n_{ik}	Manning's roughness coefficient for link ik
$NKDj$	Number of superlinks with downstream end attached to superjunction j
$NKUj$	Number of superlinks with upstream end attached to superjunction j
$NODj$	Number of orifices with downstream end attached to superjunction j
$NOUj$	Number of orifices with upstream end attached to superjunction j
$NPDj$	Number of pumps with downstream end attached to superjunction j
$NPUj$	Number of pumps with upstream end attached to superjunction j
$NWDj$	Number of weirs with downstream end attached to superjunction j
$NWUj$	Number of weirs with upstream end attached to superjunction j
q_f	Runoff rate
q_o	Lateral inflow per unit length (generic)
Q	Discharge (general)
Q_{ik}	Discharge in link ik
Q_{Ik}	Discharge at junction Ik
$Q_{in,Ik}$	External flow input at junction Ik
$Q_{in,j}$	External flow input at superjunction j
Q_o	Discharge from orifice o
Q_p	Discharge from pump p
Q_{dk}	Discharge at downstream end of superlink k
Q_{uk}	Discharge at upstream end of superlink k
Q_w	Discharge from weir w
$\hat{r}$	Estimate of truncation error
R_{ik}	Hydraulic radius for link ik
s_w	Side slope (run/rise) for triangular portion of weir w
$S_{o,ik}$	Channel bottom slope at link ik
$S_{f,ik}$	Friction head loss slope at link ik
$S_{L,ik}$	Local head loss slope at link ik
t	Time (general)
Δt	Time step
TOL	Error tolerance
$\mathbf{u}(t)$	Input vector for state-space system
u	Velocity (general)
u_{ik}	Velocity of flow at link ik
u_{Ik}	Velocity of flow at junction Ik
$U_{Ik}, V_{Ik}, W_{Ik}, T_{ik}$	Forward recurrence relation coefficients
$X_{Ik}, Y_{Ik}, Z_{Ik}, O_{ik}$	Backward recurrence relation coefficients
x	Distance (general)
$\mathbf{x}(t)$	State vector for state-space system
Δx_{ik}	Length of link ik
$y_{max,o}$	Maximum height of orifice o
$y_{max,w}$	Maximum height of weir w
$z_{inv,j}$	Invert elevation of superjunction j

Table S4.4. Glossary for hydraulic solver derivations, continued.

Variable	Description
$z_{inv,dk}$	Invert elevation at downstream end of superlink k
$z_{inv,uk}$	Invert elevation at upstream end of superlink k
$z_{inv,jdo}$	Invert elevation of superjunction at downstream end of orifice o
$z_{inv,juo}$	Invert elevation of superjunction at upstream end of orifice o
$z_{inv,jdp}$	Invert elevation of superjunction at downstream end of pump p
$z_{inv,jup}$	Invert elevation of superjunction at upstream end of pump p
$z_{inv,jdw}$	Invert elevation of superjunction at downstream end of weir w
$z_{inv,juw}$	Invert elevation of superjunction at upstream end of weir w
z_o	Offset elevation of orifice o
z_p	Offset elevation of pump p
z_w	Offset elevation of weir w
$\alpha_{uk}, \beta_{uk}, \chi_{uk}, \gamma_{uk}$	Superlink upstream flow coefficients
$\alpha_{dk}, \beta_{dk}, \chi_{dk}, \gamma_{dk}$	Superlink downstream flow coefficients
$\alpha_{uo}, \beta_{uo}, \chi_{uo}, \gamma_{uo}$	Orifice upstream flow coefficients
$\alpha_{do}, \beta_{do}, \chi_{do}, \gamma_{do}$	Orifice downstream flow coefficients
$\alpha_{up}, \beta_{up}, \chi_{up}, \gamma_{up}$	Pump upstream flow coefficients
$\alpha_{dp}, \beta_{dp}, \chi_{dp}, \gamma_{dp}$	Pump downstream flow coefficients
$\alpha_{uw}, \beta_{uw}, \chi_{uw}, \gamma_{uw}$	Weir upstream flow coefficients
$\alpha_{dw}, \beta_{dw}, \chi_{dw}, \gamma_{dw}$	Weir downstream flow coefficients
ϵ	Allowable truncation error
π_1, π_2	Intermediate coefficients
ψ_f	Suction head of wetting front
$\Psi^k, \Phi^k, F^k_{j,j}, G^k_j$	Superlink solution matrix coefficients
$\Psi^o, \Phi^o, F^o_{j,j}, G^o_j$	Orifice solution matrix coefficients
$\Psi^p, \Phi^p, F^p_{j,j}, G^p_j$	Pump solution matrix coefficients
$\Psi^w, \Phi^w, F^w_{j,j}, G^w_j$	Weir solution matrix coefficients
ζ, ξ, ρ	Filter parameters
θ_d	Soil moisture deficit
θ_{uk}	Indicator variable for submerged upstream end of superlink
θ_{dk}	Indicator variable for submerged downstream end of superlink
ω	Input signal to control structure (proportion open)

S4.2 Discretization of momentum

In the following section, we describe the discretization scheme for the momentum equation. The momentum equation is applied around each link, as shown in Figure S4.1.

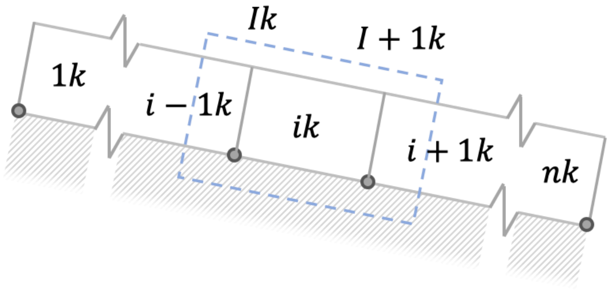


Figure S4.1. Control volume and indexing scheme for momentum equation.

Starting with the equation for conservation of momentum:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(Qu) + gA\left(\frac{\partial h}{\partial x} - S_0 + S_f + S_L\right) = 0 \tag{4.1}$$

The following discretization scheme can be applied to link ik [40]:

$$\begin{aligned} (Q_{ik}^{t+\Delta t} - Q_{ik}^{t})\frac{\Delta x_{ik}}{\Delta t} + u_{I+1k}Q_{I+1k}^{t+\Delta t} - u_{Ik}Q_{Ik}^{t+\Delta t} \\ + gA_{ik}(h_{I+1k}^{t+\Delta t} - h_{Ik}^{t+\Delta t}) - gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(S_{f,ik} + S_{L,ik})\Delta x = 0 \end{aligned} \tag{4.2}$$

This equation can be written in terms of the following coefficient equation [40]:

$$\boxed{a_{ik}Q_{i-1k}^{t+\Delta t} + b_{ik}Q_{ik}^{t+\Delta t} + c_{ik}Q_{i+1k}^{t+\Delta t} = P_{ik} + gA_{ik}(h_{Ik}^{t+\Delta t} - h_{I+1k}^{t+\Delta t})} \tag{4.3}$$

Where:

$$\boxed{a_{ik} = -\max(u_{Ik}, 0)} \tag{4.4}$$

$$\boxed{c_{ik} = -\max(-u_{I+1k}, 0)} \tag{4.5}$$

$$\boxed{b_{ik} = \frac{\Delta x_{ik}}{\Delta t} + \frac{gn_{ik}^2|Q_{ik}^t|\Delta x_{ik}}{A_{ik}R_{ik}^{4/3}} + \frac{A_{ik}|Q_{ik}^t|}{A_{cik}^2 C_{ik}^2} - a_{ik} - c_{ik}} \tag{4.6}$$

$$\boxed{P_{ik} = Q^{t}_{ik}\frac{\Delta x_{ik}}{\Delta t} + gA_{ik}S_{o,ik}\Delta x_{ik}} \tag{4.7}$$

This coefficient equation can be verified by substituting the expressions for the coefficients:

$$\begin{aligned}
&-\max(u_{Ik},0)Q^{t+\Delta t}_{i-1k}\\
+\left(\frac{\Delta x_{ik}}{\Delta t} + \frac{gn^2_{ik}|Q^t_{ik}|\Delta x_{ik}}{A_{ik}R^{4/3}_{ik}} + \frac{A_{ik}|Q^t_{ik}|}{A^2_{cik}C^2_{ik}} + \max(u_{Ik},0) + \max(-u_{I+1k},0)\right)&Q^{t+\Delta t}_{ik}\\
&-\max(-u_{I+1k},0)Q^{t+\Delta t}_{i+1k}\\
= Q_{ik}\frac{\Delta x_{ik}}{\Delta t} + gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k})&
\end{aligned} \tag{4.8}$$

Assuming $u_{ik} > 0$ and $u_{i-1k} \approx u_{ik} \approx u_{i+1k}$:

$$\begin{aligned}
-u_{Ik}Q^{t+\Delta t}_{i-1k} + \left(\frac{\Delta x_{ik}}{\Delta t} + \frac{gn^2_{ik}|Q^t_{ik}|\Delta x_{ik}}{A_{ik}R^{4/3}_{ik}} + \frac{A_{ik}|Q^t_{ik}|}{A^2_{cik}C^2_{ik}} + u_{Ik}\right)Q^{t+\Delta t}_{ik}\\
= Q^t_{ik}\frac{\Delta x_{ik}}{\Delta t} + gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k})
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
(Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{ik} - Q^{t+\Delta t}_{i-1k})u_{Ik}\\
+gA_{ik}(\frac{n^2_{ik}|Q^t_{ik}|Q^{t+\Delta t}_{ik}}{A^2_{ik}R^{4/3}_{ik}} + \frac{|Q^t_{ik}|Q^{t+\Delta t}_{ik}}{gC^2_{ik}A^2_{cik}\Delta x_{ik}})\Delta x_{ik}\\
= gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k})
\end{aligned} \tag{4.10}$$

$$\begin{aligned}
(Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{ik} - Q^{t+\Delta t}_{i-1k})u_{Ik} + gA_{ik}(S_{f,ik} + S_{L,ik})\Delta x_{ik}\\
= gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k})
\end{aligned} \tag{4.11}$$

Which further simplifies to the original combined mass and momentum balance:

$$\begin{aligned}
(Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{ik} - Q^{t+\Delta t}_{i-1k})u_{Ik}\\
+gA_{ik}(h^{t+\Delta t}_{I+1k} - h^{t+\Delta t}_{Ik}) + gA_{ik}(S_{f,ik} - gA_{ik}S_{o,ik}\Delta x_{ik} + S_{L,ik})\Delta x_{ik}
\end{aligned} \tag{4.12}$$

Alternatively, assuming $u_{ik} < 0$ and $u_{i-1k} \approx u_{ik} \approx u_{i+1k}$:

$$\begin{aligned} u_{I+1k}Q^{t+\Delta t}_{i+1k} + \left(\frac{\Delta x_{ik}}{\Delta t} + \frac{gn^2_{ik}|Q^t_{ik}|\Delta x_{ik}}{A_{ik}R^{4/3}_{ik}} + \frac{A_{ik}|Q^t_{ik}|}{A^2_{cik}C^2_{ik}} - u_{I+1k}\right)Q^{t+\Delta t}_{ik} \\ = Q^t_{ik}\frac{\Delta x_{ik}}{\Delta t} + gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k}) \end{aligned} \tag{4.13}$$

$$\begin{aligned} (Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{i+1k} - Q^{t+\Delta t}_{ik})u_{I+1k} \\ +gA_{ik}(\frac{n^2_{ik}|Q^t_{ik}|Q^{t+\Delta t}_{ik}}{A^2_{ik}R^{4/3}_{ik}} + \frac{|Q^t_{ik}|Q^{t+\Delta t}_{ik}}{gC^2_{ik}A^2_{cik}\Delta x_{ik}})\Delta x_{ik} \\ = gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k}) \end{aligned} \tag{4.14}$$

$$\begin{aligned} (Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{i+1k} - Q^{t+\Delta t}_{ik})u_{I+1k} + gA_{ik}(S_{f,ik} + S_{L,ik})\Delta x_{ik} \\ = gA_{ik}S_{o,ik}\Delta x_{ik} + gA_{ik}(h^{t+\Delta t}_{Ik} - h^{t+\Delta t}_{I+1k}) \end{aligned} \tag{4.15}$$

Which simplifies to the original combined mass and momentum balance:

$$\begin{aligned} (Q^{t+\Delta t}_{ik} - Q^t_{ik})\frac{\Delta x_{ik}}{\Delta t} + (Q^{t+\Delta t}_{i+1k} - Q^{t+\Delta t}_{ik})u_{I+1k} \\ +gA_{ik}(h^{t+\Delta t}_{I+1k} - h^{t+\Delta t}_{Ik}) + gA_{ik}(S_{f,ik} - gA_{ik}S_{o,ik}\Delta x_{ik} + S_{L,ik})\Delta x_{ik} \end{aligned} \tag{4.16}$$

S4.3 Discretization of continuity

In the following section, we describe the discretization scheme for the continuity equation. The continuity equation is applied around each junction, as shown in Figure S4.2.

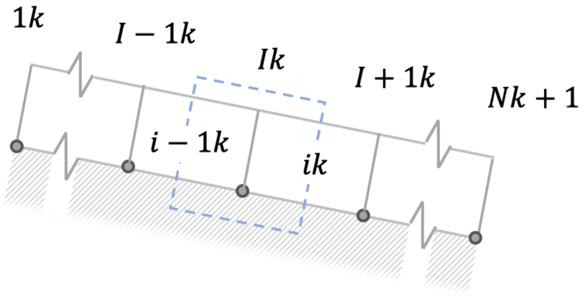


Figure S4.2. Control volume and indexing scheme for continuity equation.

Starting with the continuity equation:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_0 \tag{4.1}$$

The following discretization scheme can be applied to junction Ik [40]:

$$Q_{ik}^{t+\Delta t} - Q_{i-1k}^{t+\Delta t} + \left(\frac{B_{ik}\Delta x_{ik}}{2} + \frac{B_{i-1k}\Delta x_{i-1k}}{2} + A_{s,Ik}\right) \cdot \frac{h_{Ik}^{t+\Delta t} - h_{Ik}^{t}}{\Delta t} = Q_{in,Ik} \tag{4.2}$$

Where $B_{0k} = \Delta x_{0k} = B_{nk+1} = \Delta x_{nk+1} = 0$. Through substitution, the discretized continuity equation can be represented as follows [40]:

$$\boxed{Q_{ik}^{t+\Delta t} - Q_{i-1k}^{t+\Delta t} + E_{Ik}h_{Ik}^{t+\Delta t} = D_{Ik}} \tag{4.3}$$

Where:

$$\boxed{D_{Ik} = Q_{in,Ik} + \frac{h_{Ik}^{t}}{\Delta t}\left(\frac{B_{ik}\Delta x_{ik}}{2} + \frac{B_{i-1k}\Delta x_{i-1k}}{2} + A_{s,Ik}\right)} \tag{4.4}$$

$$\boxed{E_{Ik} = \frac{1}{\Delta t}\left(\frac{B_{ik}\Delta x_{ik}}{2} + \frac{B_{i-1k}\Delta x_{i-1k}}{2} + A_{s,Ik}\right)} \tag{4.5}$$

S4.4 Recurrence relationships

The SUPERLINK algorithm uses a series of recurrence relationships to embed channel dynamics into the solution matrix. In this section, the forward and backward recurrence relations for each superlink are derived.

Forward recurrence

Starting at the upstream end of superlink k, the continuity and momentum equations can be written as:

$$Q_{1k}^{t+\Delta t} - Q_{uk}^{t+\Delta t} + E_{1k} h_{1k}^{t+\Delta t} = D_{1k} \tag{4.1}$$

$$a_{1k} Q_{uk}^{t+\Delta t} + b_{1k} Q_{1k}^{t+\Delta t} + c_{1k} Q_{2k}^{t+\Delta t} = P_{1k} + gA_{1k}(h_{1k}^{t+\Delta t} - h_{2k}^{t+\Delta t}) \tag{4.2}$$

Combining these equations together:

$$a_{1k}(Q_{1k}^{t+\Delta t} + E_{1k} h_{1k}^{t+\Delta t} - D_{1k}) + b_{1k} Q_{1k}^{t+\Delta t} + c_{1k} Q_{2k}^{t+\Delta t} = P_{1k} + gA_{1k}(h_{1k}^{t+\Delta t} - h_{2k}^{t+\Delta t}) \tag{4.3}$$

Rearranging:

$$(a_{1k} + b_{1k}) Q_{1k}^{t+\Delta t} + c_{1k} Q_{2k}^{t+\Delta t} + (a_{1k} E_{1k} - gA_{1k}) h_{1k}^{t+\Delta t} + gA_{1k} h_{2k}^{t+\Delta t} = a_{1k} D_{1k} + P_{1k} \tag{4.4}$$

Applying continuity to the second junction:

$$Q_{2k}^{t+\Delta t} - Q_{1k}^{t+\Delta t} + E_{2k} h_{2k}^{t+\Delta t} = D_{2k} \tag{4.5}$$

Combining this equation with the previous expression:

$$\begin{aligned} (a_{1k} + b_{1k}) Q_{1k}^{t+\Delta t} + c_{1k}(Q_{1k}^{t+\Delta t} + D_{2k} - E_{2k} h_{2k}) \\ + (a_{1k} E_{1k} - gA_{1k}) h_{1k}^{t+\Delta t} + gA_{1k} h_{2k}^{t+\Delta t} = a_{1k} D_{1k} + P_{1k} \end{aligned} \tag{4.6}$$

Collecting terms:

$$\begin{aligned} (a_{1k} + b_{1k} + c_{1k}) Q_{1k}^{t+\Delta t} + (a_{1k} E_{1k} - gA_{1k}) h_{1k}^{t+\Delta t} \\ + (gA_{1k} - c_{1k} E_{2k}) h_{2k}^{t+\Delta t} + c_{1k} D_{2k} - a_{1k} D_{1k} - P_{1k} = 0 \end{aligned} \tag{4.7}$$

Thus, the flow at the first link can be expressed as:

$$Q_{1k}^{t+\Delta t} = \frac{(c_{1k}E_{2k} - gA_{1k})h_{2k}^{t+\Delta t} + (P_{1k} + a_{1k}D_{1k} - c_{1k}D_{2k}) + (gA_{1k} - a_{1k}E_{1k})h_{1k}^{t+\Delta t}}{a_{1k} + b_{1k} + c_{1k}} \tag{4.8}$$

Thus for the upstream end of superlink k, the following equation holds:

$$\boxed{Q_{1k}^{t+\Delta t} = U_{1k}h_{2k}^{t+\Delta t} + V_{1k} + W_{1k}h_{1k}^{t+\Delta t}} \tag{4.9}$$

Where:

$$\boxed{T_{1k} = a_{1k} + b_{1k} + c_{1k}} \tag{4.10}$$

$$\boxed{U_{1k} = \frac{E_{2k}c_{1k} - gA_{1k}}{T_{1k}}} \tag{4.11}$$

$$\boxed{V_{1k} = \frac{P_{1k} + D_{1k}a_{1k} - D_{2k}c_{1k}}{T_{1k}}} \tag{4.12}$$

$$\boxed{W_{1k} = \frac{gA_{1k} - E_{1k}a_{1k}}{T_{1k}}} \tag{4.13}$$

For the next element downstream, the continuity and momentum equations can be written:

$$Q_{3k}^{t+\Delta t} - Q_{2k}^{t+\Delta t} + E_{3k}h_{3k}^{t+\Delta t} = D_{3k} \tag{4.14}$$

$$a_{2k}Q_{1k}^{t+\Delta t} + b_{2k}Q_{2k}^{t+\Delta t} + c_{2k}Q_{3k}^{t+\Delta t} = P_{2k} + gA_{2k}(h_{2k}^{t+\Delta t} - h_{3k}^{t+\Delta t}) \tag{4.15}$$

Combining the two equations together and simplifying yields the following equation:

$$\begin{aligned} a_{2k}(Q_{2k}^{t+\Delta t} + E_{2k}h_{2k}^{t+\Delta t} - D_{2k}) + (b_{2k})Q_{2k}^{t+\Delta t} + c_{2k}(Q_{2k} - E_{3k}h_{3k}^{t+\Delta t} + D_{3k}) \\ -P_{2k} - gA_{2k}(h_{2k}^{t+\Delta t} - h_{3k}^{t+\Delta t}) = 0 \end{aligned} \tag{4.16}$$

$$\begin{aligned} (a_{2k} + b_{2k} + c_{2k})Q_{2k}^{t+\Delta t} + (E_{2k}a_{2k} - gA_{2k})h_{2k}^{t+\Delta t} \\ +(gA_{2k} - c_{2k}E_{3k})h_{3k}^{t+\Delta t} + (-D_{2k}a_{2k} + D_{3k}c_{2k} - P_{2k}) = 0 \end{aligned} \tag{4.17}$$

Multiplying $h_{2k}^{t+\Delta t}$ by $(U_{1k} - E_{2k})/(U_{1k} - E_{2k})$ and rearranging yields:

$$\begin{aligned}(a_{2k} + b_{2k} + c_{2k})Q_{2k}^{t+\Delta t} + \frac{(E_{2k}a_{2k} - gA_{2k})(U_{1k} - E_{2k})}{(U_{1k} - E_{2k})}h_{2k}^{t+\Delta t} \\ +(gA_{2k} - c_{2k}E_{3k})h_{3k}^{t+\Delta t} + (-P_{2k} - D_{2k}a_{2k} + D_{3k}c_{2k}) = 0\end{aligned} \tag{4.18}$$

Note that:

$$U_{1k}h_{2k}^{t+\Delta t} = (Q_{1k}^{t+\Delta t} - V_{1k} - W_{1k}h_{1k}^{t+\Delta t}) \tag{4.19}$$

$$E_{2k}h_{2k}^{t+\Delta t} = (D_{2k} - Q_{2k}^{t+\Delta t} + Q_{1k}^{t+\Delta t}) \tag{4.20}$$

Thus:

$$\begin{aligned}(a_{2k} + b_{2k} + c_{2k})Q_{2k}^{t+\Delta t} \\ +\frac{(E_{2k}a_{2k} - gA_{2k})}{(U_{1k} - E_{2k})}[(Q_{1k}^{t+\Delta t} - V_{1k} - W_{1k}h_{1k}^{t+\Delta t}) - (D_{2k} - Q_{2k}^{t+\Delta t} + Q_{1k}^{t+\Delta t})] \\ +(gA_{2k} - c_{2k}E_{3k})h_{3k}^{t+\Delta t} + (-P_{2k} - D_{2k}a_{2k} + D_{3k}c_{2k}) = 0\end{aligned} \tag{4.21}$$

Allowing $Q_{1k}^{t+\Delta t}$ to be eliminated:

$$\begin{aligned}(a_{2k} + b_{2k} + c_{2k})Q_{2k}^{t+\Delta t} + \frac{(E_{2k}a_{2k} - gA_{2k})}{U_{1k} - E_{2k}}Q_{2k}^{t+\Delta t} \\ +\frac{(E_{2k}a_{2k} - gA_{2k})(-W_{1k})}{U_{1k} - E_{2k}}h_{1k}^{t+\Delta} + (gA_{2k} - c_{2k}E_{3k})h_{3k}^{t+\Delta t} \\ +(-P_{2k} - D_{2k}a_{2k} + D_{3k}c_{2k} + (E_{2k}a_{2k} - gA_{2k})\frac{(-V_{1k} - D_{2k})}{(U_{1k} - E_{2k})}) = 0\end{aligned} \tag{4.22}$$

Rearranging:

$$\begin{aligned}\left(a_{2k} + b_{2k} + c_{2k} - \frac{gA_{2k} - E_{2k}a_{2k}}{U_{1k} - E_{2k}}\right)Q_{2k}^{t+\Delta t} \\ +\frac{(gA_{2k} - E_{2k}a_{2k})W_{1k}}{U_{1k} - E_{2k}}h_{1k}^{t+\Delta} + (gA_{2k} - c_{2k}E_{3k})h_{3k}^{t+\Delta t} \\ +\left(-P_{2k} - D_{2k}a_{2k} + D_{3k}c_{2k} + (gA_{2k} - E_{2k}a_{2k})\frac{V_{1k} + D_{2k}}{U_{1k} - E_{2k}}\right) = 0\end{aligned} \tag{4.23}$$

$$
\begin{aligned}
\left(a_{2k}+b_{2k}+c_{2k}-\frac{gA_{2k}-E_{2k}a_{2k}}{U_{1k}-E_{2k}}\right)Q_{2k}^{t+\Delta t} \\
=(E_{3k}c_{2k}-gA_{2k})h_{3k}^{t+\Delta t} \\
+\left(P_{2k}+D_{2k}a_{2k}-D_{3k}c_{2k}-(gA_{2k}-E_{2k}a_{2k})\frac{V_{1k}+D_{2k}}{(U_{1k}-E_{2k})}\right) \\
-\frac{(gA_{2k}-E_{2k}a_{2k})W_{1k}}{U_{1k}-E_{2k}}h_{1k}^{t+\Delta}
\end{aligned}
\tag{4.24}
$$

Generalizing for $i=2$, $I=2$:

$$
\begin{aligned}
\left(a_{ik}+b_{ik}+c_{ik}-\frac{gA_{ik}-E_{Ik}a_{ik}}{U_{I-1k}-E_{Ik}}\right)Q_{ik}^{t+\Delta t} \\
=(E_{I+1k}c_{ik}-gA_{ik})h_{I+1k}^{t+\Delta t} \\
+\left(P_{ik}+D_{Ik}a_{ik}-D_{I+1k}c_{ik}-(gA_{ik}-E_{ik}a_{ik})\frac{V_{I-1k}+D_{Ik}}{U_{I-1k}-E_{Ik}}\right) \\
-\frac{(gA_{ik}-E_{Ik}a_{ik})W_{I-1k}}{U_{I-1k}-E_{Ik}}h_{1k}^{t+\Delta}
\end{aligned}
\tag{4.25}
$$

Condensing in terms of the coefficients yields the following recurrence relation for the mass and momentum balance in the forward direction:

$$\boxed{Q_{ik}^{t+\Delta t}=U_{Ik}h_{I+1k}^{t+\Delta t}+V_{Ik}+W_{Ik}h_{1k}^{t+\Delta t}} \tag{4.26}$$

Where:

$$\boxed{U_{Ik}=\frac{E_{I+1k}c_{ik}-gA_{ik}}{T_{ik}}} \tag{4.27}$$

$$\boxed{V_{Ik}=\frac{P_{ik}+D_{Ik}a_{ik}-D_{I+1k}c_{ik}-(gA_{ik}-E_{Ik}a_{ik})\frac{V_{I-1k}+D_{Ik}}{U_{I-1k}-E_{Ik}}}{T_{ik}}} \tag{4.28}$$

$$\boxed{W_{Ik}=-\frac{(gA_{ik}-E_{Ik}a_{ik})W_{I-1k}}{(U_{I-1k}-E_{Ik})T_{ik}}} \tag{4.29}$$

$$\boxed{T_{ik}=\left(a_{ik}+b_{ik}+c_{ik}-\frac{gA_{ik}-E_{Ik}a_{ik}}{U_{I-1k}-E_{Ik}}\right)} \tag{4.30}$$

Backward recurrence

Starting at the downstream end of superlink k, the continuity and momentum equations can be written as:

$$Q_{dk}^{t+\Delta t} - Q_{nk}^{t+\Delta t} + E_{Nk+1} h_{Nk+1}^{t+\Delta t} = D_{Nk+1} \tag{4.31}$$

$$a_{nk} Q_{nk-1}^{t+\Delta t} + b_{nk} Q_{nk}^{t+\Delta t} + c_{nk} Q_{dk}^{t+\Delta t} = P_{nk} + gA_{nk}(h_{Nk}^{t+\Delta t} - h_{Nk+1}^{t+\Delta t}) \tag{4.32}$$

Combining these equations together:

$$a_{nk} Q_{nk-1}^{t+\Delta t} + b_{nk} Q_{nk}^{t+\Delta t} + c_{nk}(Q_{nk}^{t+\Delta t} + D_{Nk+1} - E_{Nk+1} h_{Nk+1}^{t+\Delta t}) = P_{nk} + gA_{nk}(h_{Nk}^{t+\Delta t} - h_{Nk+1}^{t+\Delta t}) \tag{4.33}$$

Rearranging:

$$(b_{nk} + c_{nk}) Q_{nk}^{t+\Delta t} + a_{nk} Q_{nk-1}^{t+\Delta t} - gA_{Nk} h_{Nk}^{t+\Delta t} + (gA_{Nk} - c_{nk} E_{Nk+1}) h_{Nk+1}^{t+\Delta t} = P_{nk} - c_{nk} D_{Nk+1} \tag{4.34}$$

Applying continuity to the penultimate junction:

$$Q_{nk}^{t+\Delta t} - Q_{nk-1}^{t+\Delta t} + E_{Nk} h_{Nk}^{t+\Delta t} = D_{Nk} \tag{4.35}$$

Combining this equation with the previous expression:

$$\begin{aligned} (b_{nk} + c_{nk}) Q_{nk}^{t+\Delta t} + a_{nk}(Q_{nk} + E_{Nk} h_{Nk}^{t+\Delta t} - D_{Nk}) \\ -gA_{Nk} h_{Nk}^{t+\Delta t} + (gA_{Nk} - c_{nk} E_{Nk+1}) h_{Nk+1}^{t+\Delta t} = P_{nk} - c_{nk} D_{Nk+1} \end{aligned} \tag{4.36}$$

Collecting terms:

$$\begin{aligned} (a_{nk} + b_{nk} + c_{nk}) Q_{nk}^{t+\Delta t} + (a_{nk} E_{Nk} - gA_{nk}) h_{Nk}^{t+\Delta t} \\ +(gA_{nk} - c_{nk} E_{Nk+1}) h_{Nk+1}^{t+\Delta t} + c_{nk} D_{Nk+1} - a_{nk} D_{Nk} - P_{nk} = 0 \end{aligned} \tag{4.37}$$

Thus, the flow at the final link can be expressed as:

$$Q_{nk}^{t+\Delta t} = \frac{(gA_{nk} - a_{nk}E_{Nk})h_{Nk}^{t+\Delta t} + (P_{nk} + a_{nk}D_{Nk} - c_{nk}D_{Nk+1})}{a_{nk} + b_{nk} + c_{nk}} + \frac{(c_{nk}E_{Nk+1} - gA_{Nk})h_{Nk+1}^{t+\Delta t}}{a_{nk} + b_{nk} + c_{nk}} \tag{4.38}$$

Thus for the downstream end of superlink k, the mass and momentum balance can be represented by the following equation:

$$\boxed{Q_{nk}^{t+\Delta t} = X_{Nk}h_{Nk}^{t+\Delta t} + Y_{Nk} + Z_{Nk}h_{Nk+1}^{t+\Delta t}} \tag{4.39}$$

Where:

$$\boxed{O_{nk} = a_{nk} + b_{nk} + c_{nk}} \tag{4.40}$$

$$\boxed{X_{Nk} = \frac{(gA_{nk} - E_{Nk}a_{nk})}{O_{nk}}} \tag{4.41}$$

$$\boxed{Y_{Nk} = \frac{P_{nk} + D_{Nk}a_{nk} - D_{Nk+1}c_{nk}}{O_{nk}}} \tag{4.42}$$

$$\boxed{Z_{Nk} = \frac{E_{Nk+1}c_{nk} - gA_{nk}}{O_{nk}}} \tag{4.43}$$

For the next element upstream, the continuity and momentum equations can be written:

$$Q_{nk-1}^{t+\Delta t} - Q_{nk-2}^{t+\Delta t} + E_{Nk-1}h_{Nk-1}^{t+\Delta t} = D_{Nk-1} \tag{4.44}$$

$$a_{nk-1}Q_{nk-2}^{t+\Delta t} + b_{nk-1}Q_{nk-1}^{t+\Delta t} + c_{nk-1}Q_{nk}^{t+\Delta t} = P_{nk-1} + gA_{nk-1}(h_{Nk-1}^{t+\Delta t} - h_{Nk}^{t+\Delta t}) \tag{4.45}$$

$$\begin{aligned} a_{nk-1}(Q_{nk-1}^{t+\Delta t} + E_{Nk-1}h_{Nk-1}^{t+\Delta t} - D_{Nk-1}) \\ +(b_{nk-1})Q_{nk-1}^{t+\Delta t} + c_{nk-1}(Q_{nk-1} - E_{Nk}h_{Nk}^{t+\Delta t} + D_{Nk}) \\ -P_{nk-1} - gA_{nk-1}(h_{Nk-1}^{t+\Delta t} - h_{Nk}^{t+\Delta t}) = 0 \end{aligned} \tag{4.46}$$

$$\begin{aligned}(a_{nk-1}+b_{nk-1}+c_{nk-1})Q_{nk-1}^{t+\Delta t}+(-E_{Nk}c_{nk-1}+gA_{nk-1})h_{Nk}^{t+\Delta t}\\+(E_{Nk-1}a_{nk-1}-gA_{nk-1})h_{Nk-1}^{t+\Delta t}+(-D_{Nk-1}a_{nk-1}+D_{Nk}c_{nk-1}-P_{nk-1})=0\end{aligned}\tag{4.47}$$

Multiplying $h_{Nk}^{t+\Delta t}$ by $(X_{Nk}+E_{Nk})/(X_{Nk}+E_{Nk})$ and rearranging:

$$\begin{aligned}(a_{nk-1}+b_{nk-1}+c_{nk-1})Q_{nk-1}^{t+\Delta t}+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})(X_{Nk}+E_{Nk})}{(X_{Nk}+E_{Nk})}h_{Nk}^{t+\Delta t}\\+(E_{Nk-1}a_{nk-1}-gA_{nk-1})h_{Nk-1}^{t+\Delta t}+(-P_{nk-1}-D_{Nk-1}a_{nk-1}+D_{Nk}c_{nk-1})=0\end{aligned}\tag{4.48}$$

Note that:

$$X_{Nk}h_{Nk}^{t+\Delta t}=(Q_{nk}^{t+\Delta t}-Y_{Nk}-Z_{Nk}h_{Nk+1}^{t+\Delta t})\tag{4.49}$$

$$E_{Nk}h_{Nk}^{t+\Delta t}=(D_{Nk}-Q_{nk}^{t+\Delta t}+Q_{nk-1}^{t+\Delta t})\tag{4.50}$$

Thus:

$$\begin{aligned}(a_{nk-1}+b_{nk-1}+c_{nk-1})Q_{nk-1}^{t+\Delta t}\\+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})}{(X_{Nk}+E_{Nk})}[(Q_{nk}^{t+\Delta t}-Y_{Nk}-Z_{Nk}h_{Nk+1}^{t+\Delta t})+(D_{Nk}-Q_{nk}^{t+\Delta t}+Q_{nk-1}^{t+\Delta t})]\\+(E_{Nk-1}a_{nk-1}-gA_{nk-1})h_{Nk-1}^{t+\Delta t}+(-P_{nk-1}-D_{Nk-1}a_{nk-1}+D_{Nk}c_{nk-1})=0\end{aligned}\tag{4.51}$$

Allowing $Q_{nk}^{t+\Delta t}$ to be eliminated:

$$\begin{aligned}(a_{nk-1}+b_{nk-1}+c_{nk-1})Q_{nk-1}^{t+\Delta t}\\+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})}{(X_{Nk}+E_{Nk})}Q_{nk-1}^{t+\Delta t}+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})(-Z_{Nk})}{(X_{Nk}+E_{Nk})}h_{Nk+1}^{t+\Delta t}\\+(E_{Nk-1}a_{nk-1}-gA_{nk-1})h_{Nk-1}^{t+\Delta t}\\+\left(-P_{nk-1}-D_{Nk-1}a_{nk-1}+D_{Nk}c_{nk-1}+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})(-Y_{Nk}+D_{Nk})}{(X_{Nk}+E_{Nk})}\right)=0\end{aligned}\tag{4.52}$$

Rearranging:

$$\begin{aligned}\left(a_{nk-1}+b_{nk-1}+c_{nk-1}+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})}{(X_{Nk}+E_{Nk})}\right)Q^{t+\Delta t}_{nk-1}\\ +(E_{Nk-1}a_{nk-1}-gA_{nk-1})h^{t+\Delta t}_{Nk-1}-\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})Z_{Nk}}{(X_{Nk}+E_{Nk})}h^{t+\Delta t}_{Nk+1}\\ +\left(-P_{nk-1}-D_{Nk-1}a_{nk-1}+D_{Nk}c_{nk-1}+(gA_{nk-1}-E_{Nk}c_{nk-1})\frac{(D_{Nk}-Y_{Nk})}{(X_{Nk}+E_{Nk})}\right)=0\end{aligned} \tag{4.53}$$

$$\begin{aligned}\left(a_{nk-1}+b_{nk-1}+c_{nk-1}+\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})}{(X_{Nk}+E_{Nk})}\right)Q^{t+\Delta t}_{nk-1}\\ =(gA_{nk-1}-E_{Nk-1}a_{nk-1})h^{t+\Delta t}_{Nk-1}\\ +\left(P_{nk-1}+D_{Nk-1}a_{nk-1}-D_{Nk}c_{nk-1}-(gA_{nk-1}-E_{Nk}c_{nk-1})\frac{(D_{Nk}-Y_{Nk})}{(X_{Nk}+E_{Nk})}\right)\\ +\frac{(gA_{nk-1}-E_{Nk}c_{nk-1})Z_{Nk}}{(X_{Nk}+E_{Nk})}h^{t+\Delta t}_{Nk+1}\end{aligned} \tag{4.54}$$

Generalizing for $i = nk - 1$, $I = Nk - 1$:

$$\begin{aligned}\left(a_{ik}+b_{ik}+c_{ik}+\frac{(gA_{ik}-E_{I+1k}c_{ik})}{(X_{I+1k}+E_{I+1k})}\right)Q^{t+\Delta t}_{ik}\\ =(gA_{ik}-E_{Ik}a_{ik})h^{t+\Delta t}_{Ik}\\ +\left(P_{ik}+D_{Ik}a_{ik}-D_{I+1k}c_{ik}-(gA_{ik}-E_{I+1k}c_{ik})\frac{(D_{I+1k}-Y_{I+1k})}{(X_{I+1k}+E_{I+1k})}\right)\\ +\frac{(gA_{ik}-E_{I+1k}c_{ik})Z_{I+1k}}{(X_{I+1k}+E_{I+1k})}h^{t+\Delta t}_{Nk+1}\end{aligned} \tag{4.55}$$

Condensing in terms of the coefficients yields the following recurrence relation for the mass and momentum balance in the backwards direction:

$$\boxed{Q^{t+\Delta t}_{ik}=X_{Ik}h^{t+\Delta t}_{Ik}+Y_{Ik}+Z_{Ik}h^{t+\Delta t}_{Nk+1}} \tag{4.56}$$

Where:

$$\boxed{X_{Ik}=\frac{gA_{ik}-E_{Ik}a_{ik}}{O_{ik}}} \tag{4.57}$$

$$\boxed{Y_{Ik} = \frac{P_{ik} + D_{Ik} a_{ik} - D_{I+1k} c_{ik} - (gA_{ik} - E_{I+1k} c_{ik}) \frac{(D_{I+1k} - Y_{I+1k})}{(X_{I+1k} + E_{I+1k})}}{O_{ik}}} \tag{4.58}$$

$$\boxed{Z_{Ik} = \frac{(gA_{ik} - E_{I+1k} c_{ik}) Z_{I+1k}}{(X_{I+1k} + E_{I+1k}) O_{ik}}} \tag{4.59}$$

$$\boxed{O_{ik} = \left(a_{ik} + b_{ik} + c_{ik} + \frac{gA_{ik} - E_{I+1k} c_{ik}}{X_{I+1k} + E_{I+1k}} \right)} \tag{4.60}$$

S4.5 Inlet hydraulics

Depth at upstream end of superlink

The discharge at the upstream end of a superlink is given by:

$$Q_{uk} = C_{uk} A_{uk} \sqrt{2g \Delta H_{uk}} \tag{4.1}$$

Where:

$$\Delta H_{uk} = \theta_{uk}(H_{juk} - z_{inv,uk}) - h_{uk} \tag{4.2}$$

And:

$$\theta_{uk} = \begin{cases} 1 & H_{juk} \geqslant z_{inv,uk} \\ 0 & H_{juk} < z_{inv,uk} \end{cases} \tag{4.3}$$

Squaring and rearranging provides the depth boundary condition at the upstream end [137]:

$$Q_{uk}^2 = 2C_{uk}^2 A_{uk}^2 g(\theta_{uk}[H_{juk} - z_{inv,uk}] - h_{uk}) \tag{4.4}$$

$$|Q_{uk}^t| Q_{uk}^{t+\Delta t} = 2C_{uk}^2 A_{uk}^2 g(\theta_{uk}[H_{juk} - z_{inv,uk}] - h_{uk}) \tag{4.5}$$

$$\boxed{h_{uk} = -\frac{|Q_{uk}^t| Q_{uk}^{t+\Delta t}}{2C_{uk}^2 A_{uk}^2 g} + \theta_{uk}(H_{juk} - z_{inv,uk})} \tag{4.6}$$

Depth at downstream end of superlink

The discharge at the downstream end of a superlink is given by:

$$Q_{dk} = C_{dk} A_{dk} \sqrt{2g \Delta H_{dk}} \tag{4.7}$$

Where:

$$\Delta H_{dk} = h_{dk} + \theta_{dk}(z_{inv,dk} - H_{jdk}) \tag{4.8}$$

And:

$$\theta_{dk} = \begin{cases} 1 & H_{jdk} \geqslant z_{inv,dk} \\ 0 & H_{jdk} < z_{inv,dk} \end{cases} \tag{4.9}$$

Squaring and rearranging provides the depth boundary condition at the downstream end [137]:

$$Q_{dk}^2 = 2C_{dk}^2 A_{dk}^2 g(h_{dk} + \theta_{dk}[z_{inv,dk} - H_{jdk}]) \tag{4.10}$$

$$|Q_{dk}^t| Q_{dk}^{t+\Delta t} = 2C_{dk}^2 A_{dk}^2 g(h_{dk} + \theta_{dk}[z_{inv,dk} - H_{jdk}]) \tag{4.11}$$

$$\boxed{h_{dk} = \frac{|Q_{dk}^t| Q_{dk}^{t+\Delta t}}{2C_{dk}^2 A_{dk}^2 g} + \theta_{dk}(H_{jdk} - z_{inv,dk})} \tag{4.12}$$

Superlink boundary conditions

From the continuity equation, the inlet and outlet flows for each superlink (Q_{uk} and Q_{dk}) are defined as:

$$Q_{uk}^{t+\Delta t} = Q_{1k}^{t+\Delta t} + E_{1k} h_{1k}^{t+\Delta t} - D_{1k} \tag{4.13}$$

$$Q_{dk}^{t+\Delta t} = Q_{nk}^{t+\Delta t} - E_{Nk+1} h_{Nk+1}^{t+\Delta t} + D_{Nk+1} \tag{4.14}$$

From the recurrence relations developed in Section S4.4, it can be shown that:

$$Q_{1k}^{t+\Delta t} = X_{1k} h_{uk}^{t+\Delta t} + Y_{1k} + Z_{1k} h_{dk}^{t+\Delta t} \tag{4.15}$$

$$Q_{nk}^{t+\Delta t} = U_{Nk} h_{dk}^{t+\Delta t} + V_{Nk} + W_{Nk} h_{uk}^{t+\Delta t} \tag{4.16}$$

Thus, the inlet and outlet flows can be redefined in terms of the recurrence relations:

$$Q_{uk}^{t+\Delta t} = (X_{1k} + E_{1k}) h_{1k}^{t+\Delta t} + (Y_{1k} - D_{1k}) + Z_{1k} h_{Nk+1}^{t+\Delta t} \tag{4.17}$$

$$Q_{dk}^{t+\Delta t} = (U_{Nk} - E_{Nk+1}) h_{Nk+1}^{t+\Delta t} + (V_{Nk} + D_{Nk+1}) + W_{Nk} h_{1k}^{t+\Delta t} \tag{4.18}$$

To simplify the algebra, define the following coefficients:

$$\tilde{X}_{1k} = X_{1k} + E_{1k} \tag{4.19}$$

$$\tilde{Y}_{1k} = Y_{1k} - D_{1k} \tag{4.20}$$

$$\tilde{U}_{Nk} = U_{Nk} - E_{Nk+1} \tag{4.21}$$

$$\tilde{V}_{Nk} = V_{Nk} + D_{Nk+1} \tag{4.22}$$

$$\tag{4.23}$$

Thus, the inlet and outlet flows can be redefined as follows:

$$Q_{uk}^{t+\Delta t} = \tilde{X}_{1k} h_{1k}^{t+\Delta t} + \tilde{Y}_{1k} + Z_{1k} h_{Nk+1}^{t+\Delta t} \tag{4.24}$$

$$Q_{dk}^{t+\Delta t} = \tilde{U}_{Nk} h_{Nk+1}^{t+\Delta t} + \tilde{V}_{Nk} + W_{Nk} h_{1k}^{t+\Delta t} \tag{4.25}$$

From the depth boundary conditions at the ends of each superlink, it is known that:

$$h_{1k} = \gamma_{uk} Q_{uk}^{t+\Delta t} + \theta_{uk}(H_{juk} - z_{inv,uk}) \tag{4.26}$$

$$h_{Nk+1} = \gamma_{dk} Q_{dk}^{t+\Delta t} + \theta_{dk}(H_{jdk} - z_{inv,dk}) \tag{4.27}$$

Where:

$$\gamma_{uk} = -\frac{|Q_{uk}^{t}|}{2C_{uk}^{2} A_{uk}^{2} g} \tag{4.28}$$

$$\gamma_{dk} = \frac{|Q_{dk}^{t}|}{2C_{dk}^{2} A_{dk}^{2} g} \tag{4.29}$$

Substituting the boundary conditions into the recurrence relations yields the following equations:

$$Q_{uk}^{t+\Delta t} = \tilde{X}_{1k}(\gamma_{uk} Q_{uk}^{t+\Delta t} + \theta_{uk}[H_{juk} - z_{inv,uk}]) + \tilde{Y}_{1k} + Z_{1k}(\gamma_{dk} Q_{dk}^{t+\Delta t} + \theta_{dk}[H_{jdk} - z_{inv,dk}]) \tag{4.30}$$

$$Q_{dk}^{t+\Delta t} = \tilde{U}_{Nk}(\gamma_{dk} Q_{dk}^{t+\Delta t} + \theta_{dk}[H_{jdk} - z_{inv,dk}]) + \tilde{V}_{Nk} + W_{Nk}(\gamma_{uk} Q_{uk}^{t+\Delta t} + \theta_{uk}[H_{juk} - z_{inv,uk}]) \tag{4.31}$$

Expanding:

$$\begin{aligned} Q_{uk}^{t+\Delta t} = \tilde{X}_{1k}\gamma_{uk}Q_{uk}^{t+\Delta t} + \tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} - \tilde{X}_{1k}\theta_{uk}z_{inv,uk} + \tilde{Y}_{1k} \\ +Z_{1k}\gamma_{dk}Q_{dk}^{t+\Delta t} + Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} - Z_{1k}\theta_{dk}z_{inv,dk} \end{aligned} \tag{4.32}$$

$$\begin{aligned} Q_{dk}^{t+\Delta t} = \tilde{U}_{Nk}\gamma_{dk}Q_{dk}^{t+\Delta t} + \tilde{U}_{Nk}\theta_{dk}H_{jdk} - \tilde{U}_{Nk}\theta_{dk}z_{inv,dk} + \tilde{V}_{Nk} \\ +W_{Nk}\gamma_{uk}Q_{uk}^{t+\Delta t} + W_{Nk}\theta_{uk}H_{juk} - W_{Nk}\theta_{uk}z_{inv,uk} \end{aligned} \tag{4.33}$$

Rearranging:

$$0 = (\tilde{X}_{1k}\gamma_{uk} - 1)Q_{uk}^{t+\Delta t} + Z_{1k}\gamma_{dk}Q_{dk}^{t+\Delta t} + \tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} + Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} + \pi_1 \tag{4.34}$$

$$0 = W_{Nk}\gamma_{uk}Q_{uk}^{t+\Delta t} + (\tilde{U}_{Nk}\gamma_{dk} - 1)Q_{dk}^{t+\Delta t} + W_{Nk}\theta_{uk}H_{juk} + \tilde{U}_{Nk}\theta_{dk}H_{jdk} + \pi_2 \tag{4.35}$$

Where:

$$\pi_1 = \tilde{Y}_{1k} - \tilde{X}_{1k}\theta_{uk}z_{inv,uk} - Z_{1k}\theta_{dk}z_{inv,dk} \tag{4.36}$$

$$\pi_2 = \tilde{V}_{Nk} - W_{Nk}\theta_{uk}z_{inv,uk} - \tilde{U}_{Nk}\theta_{dk}z_{inv,dk} \tag{4.37}$$

Writing as a matrix equation:

$$\begin{bmatrix} (\tilde{X}_{1k}\gamma_{uk} - 1) & Z_{1k}\gamma_{dk} \\ W_{Nk}\gamma_{uk} & (\tilde{U}_{Nk}\gamma_{dk} - 1) \end{bmatrix} \begin{bmatrix} Q_{uk}^{t+\Delta t} \\ Q_{dk}^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} -\tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} - Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_1 \\ -W_{Nk}\theta_{uk}H_{juk}^{t+\Delta t} - \tilde{U}_{Nk}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_2 \end{bmatrix} \tag{4.38}$$

Taking the matrix inverse:

$$\begin{bmatrix} Q_{uk}^{t+\Delta t} \\ Q_{dk}^{t+\Delta t} \end{bmatrix} = \frac{1}{D_k^*} \begin{bmatrix} (\tilde{U}_{Nk}\gamma_{dk} - 1) & -Z_{1k}\gamma_{dk} \\ -W_{Nk}\gamma_{uk} & (\tilde{X}_{1k}\gamma_{uk} - 1) \end{bmatrix} \begin{bmatrix} -\tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} - Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_1 \\ -W_{Nk}\theta_{uk}H_{juk}^{t+\Delta t} - \tilde{U}_{Nk}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_2 \end{bmatrix} \tag{4.39}$$

Where:

$$D_k^* = (\tilde{X}_{1k}\gamma_{uk} - 1)(\tilde{U}_{Nk}\gamma_{dk} - 1) - (Z_{1k}\gamma_{dk})(W_{Nk}\gamma_{uk}) \tag{4.40}$$

Expanding:

$$\begin{aligned} Q_{uk}^{t+\Delta t} = &\{\tilde{U}_{Nk}\gamma_{dk} - 1)(-\tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} - Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_1) \\ &+(-Z_{1k}\gamma_{dk})(-W_{Nk}\theta_{uk}H_{juk}^{t+\Delta t} - \tilde{U}_{Nk}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_2)\}/D_k^* \end{aligned} \tag{4.41}$$

$$\begin{aligned} Q_{dk}^{t+\Delta t} = &\{(-W_{Nk}\gamma_{uk})(-\tilde{X}_{1k}\theta_{uk}H_{juk}^{t+\Delta t} - Z_{1k}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_1) \\ &+(\tilde{X}_{1k}\gamma_{uk} - 1)(-W_{Nk}\theta_{uk}H_{juk}^{t+\Delta t} - \tilde{U}_{Nk}\theta_{dk}H_{jdk}^{t+\Delta t} - \pi_2)\}/D_k^* \end{aligned} \tag{4.42}$$

Arranging in terms of the unknown heads:

$$\begin{aligned} Q_{uk}^{t+\Delta t} = &\{\theta_{uk}[(\tilde{U}_{Nk}\gamma_{dk} - 1)(-\tilde{X}_{1k}) + (-Z_{1k}\gamma_{dk})(-W_{Nk})]H_{juk}^{t+\Delta t} + \\ &\theta_{dk}[(\tilde{U}_{Nk}\gamma_{dk} - 1)(-Z_{1k}) + (-Z_{1k}\gamma_{dk})(-\tilde{U}_{Nk})]H_{jdk}^{t+\Delta t} + \\ &[(\tilde{U}_{Nk}\gamma_{dk} - 1)(-\pi_1) + (-Z_{1k}\gamma_{dk})(-\pi_2)]\}/D_k^* \end{aligned} \tag{4.43}$$

$$\begin{aligned} Q_{dk}^{t+\Delta t} = &\{\theta_{uk}[(-W_{Nk}\gamma_{uk})(-\tilde{X}_{1k}) + (\tilde{X}_{1k}\gamma_{uk} - 1)(-W_{Nk})]H_{juk}^{t+\Delta t} + \\ &\theta_{dk}[(-W_{Nk}\gamma_{uk})(-Z_{1k}) + (\tilde{X}_{1k}\gamma_{uk} - 1)(-\tilde{U}_{Nk})]H_{jdk}^{t+\Delta t} + \\ &[(-W_{Nk}\gamma_{uk})(-\pi_1) + (\tilde{X}_{1k}\gamma_{uk} - 1)(-\pi_2)]\}/D_k^* \end{aligned} \tag{4.44}$$

Finally, the flow rates at the upstream and downstream ends of superlink k can be expressed as:

$$\boxed{Q_{uk}^{t+\Delta t} = \alpha_{uk}H_{juk}^{t+\Delta t} + \beta_{uk}H_{jdk}^{t+\Delta t} + \chi_{uk}} \tag{4.45}$$

$$\boxed{Q_{dk}^{t+\Delta t} = \alpha_{dk}H_{juk}^{t+\Delta t} + \beta_{dk}H_{jdk}^{t+\Delta t} + \chi_{dk}} \tag{4.46}$$

Where:

$$\boxed{\alpha_{uk} = \frac{\theta_{uk}[(1 - (U_{Nk} - E_{Nk+1})\gamma_{dk})(X_{1k} + E_{1k}) + Z_{1k}\gamma_{dk}W_{Nk}]}{D_k^*}} \tag{4.47}$$

$$\boxed{\beta_{uk} = \frac{\theta_{dk}[(1 - (U_{Nk} - E_{Nk+1})\gamma_{dk})Z_{1k} + Z_{1k}\gamma_{dk}(U_{Nk} - E_{Nk+1})]}{D_k^*}} \tag{4.48}$$

$$\boxed{\begin{aligned}\chi_{uk} = &\frac{(1-(U_{Nk}-E_{Nk+1})\gamma_{dk})(Y_{1k}-D_{1k}-(X_{1k}+E_{1k})\theta_{uk}z_{inv,uk}-Z_{1k}\theta_{dk}z_{inv,dk})}{D_k^*}\\ &+\frac{(Z_{1k}\gamma_{dk})(V_{Nk}+D_{Nk+1}-W_{Nk}\theta_{uk}z_{inv,uk}-(U_{Nk}-E_{Nk+1})\theta_{dk}z_{inv,dk})}{D_k^*}\end{aligned}} \tag{4.49}$$

$$\boxed{\alpha_{dk} = \frac{\theta_{uk}[(1-(X_{1k}+E_{1k})\gamma_{uk})W_{Nk}+W_{Nk}\gamma_{uk}(X_{1k}+E_{1k})]}{D_k^*}} \tag{4.50}$$

$$\boxed{\beta_{dk} = \frac{\theta_{dk}[(1-(X_{1k}+E_{1k})\gamma_{uk})(U_{Nk}-E_{Nk+1})+W_{Nk}\gamma_{uk}Z_{1k}]}{D_k^*}} \tag{4.51}$$

$$\boxed{\begin{aligned}\chi_{dk} = &\frac{(1-(X_{1k}+E_{1k})\gamma_{uk})(V_{Nk}+D_{Nk+1}-W_{Nk}\theta_{uk}z_{inv,uk}-(U_{Nk}-E_{Nk+1})\theta_{dk}z_{inv,dk})}{D_k^*}\\ &+\frac{(W_{Nk}\gamma_{uk})(Y_{1k}-D_{1k}-(X_{1k}+E_{1k})\theta_{uk}z_{inv,uk}-Z_{1k}\theta_{dk}z_{inv,dk})}{D_k^*}\end{aligned}} \tag{4.52}$$

$$\boxed{D_k^* = ((X_{1k}+E_{1k})\gamma_{uk}-1)((U_{Nk}-E_{Nk+1})\gamma_{dk}-1)-(Z_{1k}\gamma_{dk})(W_{Nk}\gamma_{uk})} \tag{4.53}$$

$$\boxed{\gamma_{uk} = -\frac{|Q_{uk}^t|}{2C_{uk}^2A_{uk}^2g}} \tag{4.54}$$

$$\boxed{\gamma_{dk} = \frac{|Q_{dk}^t|}{2C_{dk}^2A_{dk}^2g}} \tag{4.55}$$

S4.6 Forming the solution matrix

The equations for the flows at the ends of each superlink are given by [40]:

$$\sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} - \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t} + Q_{in,j}^{t+\Delta t} = \frac{A_{sj}(H_j^{t+\Delta t}-H_j^t)}{\Delta t} \tag{4.1}$$

Substituting the linear expressions for the upstream and downstream flows:

$$\begin{aligned}\frac{A_{sj}(H_j^{t+\Delta t}-H_j^t)}{\Delta t} &= \sum_{l=1}^{NKDj}(\alpha_{dk_l}H_{juk_l}^{t+\Delta t}+\beta_{dk_l}H_{jdk_l}^{t+\Delta t}+\chi_{dk_l}) \\ &\quad -\sum_{m=1}^{NKUj}(\alpha_{uk_m}H_{juk_m}^{t+\Delta t}+\beta_{uk_m}H_{jdk_m}^{t+\Delta t}+\chi_{uk_m})+Q_{in,j}\end{aligned} \tag{4.2}$$

Because $H_{jdk_l}=H_j$ and $H_{juk_m}=H_j$:

$$\begin{aligned}\frac{A_{sj}(H_j^{t+\Delta t}-H_j^t)}{\Delta t} &= \sum_{l=1}^{NKDj}(\alpha_{dk_l}H_{juk_l}^{t+\Delta t}+\beta_{dk_l}H_j^{t+\Delta t}+\chi_{dk_l}) \\ &\quad -\sum_{m=1}^{NKUj}(\alpha_{uk_m}H_j^{t+\Delta t}+\beta_{uk_m}H_{jdk_m}^{t+\Delta t}+\chi_{uk_m})+Q_{in,j}\end{aligned} \tag{4.3}$$

Rearranging:

$$\begin{aligned}\left(\frac{A_{sj}}{\Delta t}+\sum_{m=1}^{NKUj}\alpha_{uk_m}-\sum_{l=1}^{NKDj}\beta_{dk_l}\right)H_j^{t+\Delta t}-\sum_{l=1}^{NKDj}\alpha_{dk_l}H_{juk_l}^{t+\Delta t}+\sum_{m=1}^{NKUj}\beta_{uk_m}H_{jdk_m}^{t+\Delta t} \\ =\frac{A_{sj}(H_j^t)}{\Delta t}+\sum_{l=1}^{NKDj}\chi_{dk_l}-\sum_{m=1}^{NKUj}\chi_{uk_m}+Q_{in,j}\end{aligned} \tag{4.4}$$

The continuity equation for each superjunction can thus be redefined in terms of the following coefficients.

$$\boxed{\begin{aligned}\left(\frac{A_{sj}}{\Delta t}+F_{j,j}^k\right)\cdot H_j^{t+\Delta t}+\sum_{\ell=1}^{NKDj}\Phi_{j,juk_\ell}^k H_{juk_\ell}^{t+\Delta t}+\sum_{m=1}^{NKUj}\Psi_{j,jdk_m}^k H_{jdk_m}^{t+\Delta t} \\ =\frac{A_{sj}}{\Delta t}H_j^t+Q_{in,j}^{t+\Delta t}+G_j^k\end{aligned}} \tag{4.5}$$

Where:

$$\boxed{F_{j,j}^k=\sum_{m=1}^{NKUj}\alpha_{uk_m}-\sum_{\ell=1}^{NKDj}\beta_{dk_\ell}} \tag{4.6}$$

$$\boxed{\Phi_{j,juk_\ell}^k=-\alpha_{dk_\ell}} \tag{4.7}$$

$$\boxed{\Psi^{k}_{j,jdk_m} = \beta_{uk_m}} \tag{4.8}$$

$$\boxed{G^{k}_{j} = \sum_{\ell=1}^{NKDj} \chi_{dk_\ell} - \sum_{m=1}^{NKUj} \chi_{uk_m}} \tag{4.9}$$

Solution matrix equation for example network

Let the sparse matrix equation be expressed as the LTV system:

$$(A_K(t) + A_S(t)) \cdot \mathbf{x}(t + \Delta t) = A_S(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + D(t) \tag{4.10}$$

Combining terms together yields the simple linear system of equations:

$$Ax = b \tag{4.11}$$

For brevity, define $\tilde{F}_{j,j} = \frac{A_{sj}}{\Delta t} + F^{k}_{j,j}$ and $\tilde{G}_j = \frac{A_{sj}}{\Delta t} H^{t}_{j} + Q_{in,j} + G^{k}_{j}$. Thus, for the example network in Ji (1998), the sparse matrix equation at time $t + \Delta t$ is expressed as:

$$\begin{bmatrix} \tilde{F}_{1,1} & \Psi^{k}_{1,2} & 0 & 0 & 0 & 0 \\ \Phi^{k}_{2,1} & \tilde{F}_{2,2} & \Psi^{k}_{2,3} & 0 & \Psi^{k}_{2,5} & 0 \\ 0 & \Phi^{k}_{3,2} & \tilde{F}_{3,3} & \Psi^{k}_{3,4} & \Phi^{k}_{3,5} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \Phi^{k}_{5,2} & \Psi^{k}_{5,3} & 0 & \tilde{F}_{5,5} & \Psi^{k}_{5,6} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H^{t+\Delta t}_{1} \\ H^{t+\Delta t}_{2} \\ H^{t+\Delta t}_{3} \\ H^{t+\Delta t}_{4} \\ H^{t+\Delta t}_{5} \\ H^{t+\Delta t}_{6} \end{bmatrix} = \begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \\ H^{t+\Delta t}_{bc,4} \\ \tilde{G}_5 \\ H^{t+\Delta t}_{bc,6} \end{bmatrix} \tag{4.12}$$

S4.7 Solving the internal superlink states

After computing the superjunction heads, the depths and flows inside each superlink are computed by combining the boundary condition equations with the superlink recurrence relations.

First the boundary condition equations at the upstream and downstream ends of each superlink are solved to determine the superlink boundary depths and flows. For the boundary flows:

$$Q_{uk}^{t+\Delta t} = \alpha_{uk} H_{juk}^{t+\Delta t} + \beta_{uk} H_{jdk}^{t+\Delta t} + \chi_{uk} \tag{4.1}$$

$$Q_{dk}^{t+\Delta t} = \alpha_{dk} H_{juk}^{t+\Delta t} + \beta_{dk} H_{jdk}^{t+\Delta t} + \chi_{dk} \tag{4.2}$$

For the boundary depths:

$$h_{1k} = \gamma_{uk} Q_{uk}^{t+\Delta t} + H_{juk} - z_{inv,uk} \tag{4.3}$$

$$h_{Nk+1} = \gamma_{dk} Q_{dk}^{t+\Delta t} + H_{jdk} - z_{inv,dk} \tag{4.4}$$

With these boundary conditions established, the internal depth states ($h_{2k}, \cdots h_{Nk}$) and flow states ($Q_{1k} \cdots Q_{Nk}$) can be solved using the forward and backward recurrence relations. From Section S4.4, the recurrence relations give the following two sets of equations.

$$Q_{ik}^{t+\Delta t} = U_{Ik} h_{I+1k}^{t+\Delta t} + V_{Ik} + W_{Ik} h_{1k}^{t+\Delta t}, \;\; i = I = 1, 2 \cdots Nk \tag{4.5}$$

$$Q_{ik}^{t+\Delta t} = X_{Ik} h_{Ik}^{t+\Delta t} + Y_{Ik} + Z_{Ik} h_{Nk+1}^{t+\Delta t}, \;\; i = I = 1, 2 \cdots Nk \tag{4.6}$$

Subtracting the second set of equations from the first:

$$0 = U_{Ik} h_{I+1k}^{t+\Delta t} - X_{Ik} h_{Ik}^{t+\Delta t} + (V_{Ik} - Y_{Ik} + W_{Ik} h_{1k}^{t+\Delta t} - Z_{Ik} h_{Nk+1}^{t+\Delta t}), \;\; I = 1, 2 \cdots Nk \tag{4.7}$$

Because $h_{1k}^{t+\Delta t}$ and $h_{Nk+1}^{t+\Delta t}$ are known, the result is a system of Nk equations with $Nk-1$ unknowns. Thus, it would appear that the system of equations is overdetermined. However, it can be shown that only $Nk - 1$ of the equations are linearly independent. A demonstration of this fact is too long to include here, but is included as a notebook in the software repository linked with the main paper.

S4.8 Addition of dynamic control structures

One major innovation we present with respect to the hydraulic model is the inclusion of dynamic control structures, such as orifices, weirs and pumps. These control structures are an integral component of many real-world stormwater networks, and thus representing their dynamics is important for achieving accurate simulation results. In addition to allowing better representation of real-world stormwater networks, our contribution paves the way for real-time control in an operational context by allowing the user to specify an input signal $\omega(t)$ for each control structure. Users may execute arbitrary dynamic control strategies, ranging from simple rule-based control to more sophisticated strategies such as linear-quadratic regulation (LQR) or model-predictive control (MPC). When combined with the Kalman filter developed in the following sections, this capability yields a powerful tool for implementing real-time control of stormwater and sewer networks. In the following section, we present the governing equations for each control structure and show how these equations can be embedded into the state space model.

S4.9 Representing orifices

For orifices, six different flow cases are possible:

- Side-mounted orifice with both sides submerged
- Side-mounted orifice with one side submerged
- Side-mounted orifice with weir-like flow
- Bottom-mounted orifice with both sides submerged
- Bottom-mounted orifice with one side submerged
- No-flow condition

Governing equations for orifices

The governing equations for each condition are presented here:

Side-mounted orifice with both sides submerged

This flow regime occurs when both of the following conditions are met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o + \omega y_{max,o}$
- $\min(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o + \frac{\omega y_{max,o}}{2}$

The effective head is computed as:

$$H_{e,o} = |H_{juo} - H_{jdo}| \tag{4.1}$$

And the flow is computed as:

$$Q_o = \text{sgn}(H_{juo} - H_{jdo}) \cdot C_o A_o \sqrt{2gH_{e,o}} \tag{4.2}$$

Side-mounted orifice with one side submerged

This flow regime occurs when both of the following conditions are met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o + \omega y_{max,o}$
- $\min(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) < z_o + \frac{\omega y_{max,o}}{2}$

The effective head is computed as:

$$H_{e,o} = \cdot \left[\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,jdo}) - (z_o + \frac{\omega y_{max,o}}{2}) \right] \tag{4.3}$$

And the flow is computed as:

$$Q_o = \text{sgn}(H_{juo} - H_{jdo}) \cdot C_o A_o \sqrt{2gH_{e,o}} \tag{4.4}$$

Side-mounted orifice with weir-like flow

This flow regime occurs when both of the following conditions are met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o$
- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) < z_o + \omega y_{max,o}$

The effective head is computed as:

$$H_{e,o} = \max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) - z_o \tag{4.5}$$

And the flow is computed as:

$$Q_o = \text{sgn}(H_{juo} - H_{jdo}) \frac{C_o A_o \sqrt{g}}{\omega y_{max,o}} \sqrt{H_{e,o}} \tag{4.6}$$

Bottom-mounted orifice with both sides submerged

This flow regime occurs when both of the following conditions are met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o$
- $\min(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o$

The effective head is computed as:

$$H_{e,o} = |H_{juo} - H_{jdo}| \tag{4.7}$$

And the flow is computed as:

$$Q_o = \mathsf{sgn}(H_{juo} - H_{jdo}) \cdot C_o A_o \sqrt{2gH_{e,o}} \tag{4.8}$$

Bottom-mounted orifice with one side submerged

This flow regime occurs when both of the following conditions are met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) > z_o$
- $\min(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) < z_o$

The effective head is computed as:

$$H_{e,o} = \cdot \left[\max(H_{uo} - z_{inv,uo}, H_{do} - z_{inv,do}) - z_o\right] \tag{4.9}$$

And the flow is computed as:

$$Q_o = \mathsf{sgn}(H_{juo} - H_{jdo}) \cdot C_o A_o \sqrt{2gH_{e,o}} \tag{4.10}$$

No-flow condition

This flow regime occurs when the following condition is met:

- $\max(H_{juo} - z_{inv,juo}, H_{jdo} - z_{inv,juo}) \leqslant z_o$

In this case, the effective head and flow are both zero:

$$H_{e,o} = 0 \tag{4.11}$$

$$Q_o = 0 \tag{4.12}$$

Representing orifice equations in the solution matrix

Orifices can be represented in the solution matrix as follows.

Define the following indicator functions:

$$\Omega(H_{juo}, H_{jdo}) = \begin{cases} 1, & H_{juo} \geqslant H_{jdo} \\ 0, & o/w \end{cases} \tag{4.13}$$

$$\tau(o) = \begin{cases} 1, & \text{orifice } o \text{ is side-mounted} \\ 0, & \text{orifice } o \text{ is bottom-mounted} \end{cases} \tag{4.14}$$

Similarly, define boolean-valued functions to represent the following flow conditions:

Submerged on high-head side

$$\Theta_{o,1} = \begin{cases} 1, & \Omega H_{juo} + (1-\Omega)H_{jdo} > z_o + z_{inv,juo} + \tau\omega y_{max,o} \\ 0, & o/w \end{cases} \tag{4.15}$$

Submerged on low-head side

$$\Theta_{o,2} = \begin{cases} 1, & (1-\Omega)H_{juo} + \Omega H_{jdo} > z_o + z_{inv,juo} + \frac{\tau\omega y_{max,o}}{2} \\ 0, & o/w \end{cases} \tag{4.16}$$

Above bottom rim on high-head side

$$\Theta_{o,3} = \begin{cases} 1, & \Omega H_{juo} + (1-\Omega)H_{jdo} > z_o + z_{inv,juo} \\ 0, & o/w \end{cases} \tag{4.17}$$

The flow through an orifice can now be represented using the following linearized coefficient equation:

$$\boxed{Q_o^{t+\Delta t} = \alpha_o H_{juo}^{t+\Delta t} + \beta_o H_{jdo}^{t+\Delta t} + \chi_o} \tag{4.18}$$

Where:

$$\boxed{\alpha_o = \begin{cases} \gamma_o \omega^2, & \Theta_{o,1} \wedge \Theta_{o,2} \\ \gamma_o \Omega (-1)^{1-\Omega} \omega^2, & \Theta_{o,1} \wedge \neg\Theta_{o,2} \\ \frac{\gamma_o}{2y_{max,o}^2} \Omega (-1)^{1-\Omega}, & \neg\Theta_{o,1} \wedge \Theta_{o,3} \\ 0, & \neg\Theta_{o,3} \end{cases}} \tag{4.19}$$

$$\boxed{\beta_o = \begin{cases} -\gamma_o \omega^2, & \Theta_{o,1} \wedge \Theta_{o,2} \\ \gamma_o (1-\Omega)(-1)^{1-\Omega} \omega^2, & \Theta_{o,1} \wedge \neg\Theta_{o,2} \\ \frac{\gamma_o}{2y_{max,o}^2} (1-\Omega)(-1)^{1-\Omega}, & \neg\Theta_{o,1} \wedge \Theta_{o,3} \\ 0, & \neg\Theta_{o,3} \end{cases}} \tag{4.20}$$

$$\boxed{\chi_o = \begin{cases} 0, & \Theta_{o,1} \wedge \Theta_{o,2} \\ \gamma_o (-1)^{1-\Omega} (-z_{inv,uo} - z_o - \frac{\tau \omega y_{max,o}}{2}), & \Theta_{o,1} \wedge \neg\Theta_{o,2} u^2 \\ \frac{\gamma_o}{2y_{max,o}^2} (-z_{inv,uo} - z_o), & \neg\Theta_{o,1} \wedge \Theta_{o,3} \\ 0, & \neg\Theta_{o,3} \end{cases}} \tag{4.21}$$

$$\boxed{\gamma_o = \frac{2gC_o^2 A_o^2}{|Q_o^t|}} \tag{4.22}$$

These equations can be added to the solution matrix in much the same way as the linearized superlink coefficients ($\alpha_{uk}, \beta_{uk}, \chi_{uk}, \alpha_{dk}, \beta_{dk}, \chi_{dk}$). Thus, define the matrix $A_O(t)$ and the vector $D_O(t)$. The elements of $A_O(t)$ are defined such that $A_{O_{j,j}}(t) = F_{j,j}^o$; $A_{O_{j,juo_\ell}} = \Phi_{j,juo_\ell}^o$; and $A_{O_{j,jdo_m}} = \Psi_{j,jdo_m}^o$ (with all other elements of $A_O(t)$ equal to zero).

$$\boxed{F_{j,j}^o = \sum_{m=1}^{NOUj} \alpha_{o_m} - \sum_{\ell=1}^{NODj} \beta_{o_\ell}} \tag{4.23}$$

$$\boxed{\Phi^{o}_{j,juo_\ell} = -\alpha_{o_\ell}} \tag{4.24}$$

$$\boxed{\Psi^{o}_{j,jdo_m} = \beta_{o_m}} \tag{4.25}$$

Similarly, the elements of $D_O(t)$ are defined such that $D_{O_j} = G^o_j$.

$$\boxed{G^o_j = \sum_{m=1}^{NOUj} \chi_{o_m} - \sum_{\ell=1}^{NODj} \chi_{o_\ell}} \tag{4.26}$$

S4.10 Representing weirs

This section discusses the governing equations for weirs, and explains how weirs can be incorporated into the solution matrix. Only transverse weirs will be considered.

Governing equations for weirs

First, without loss of generality, assume all weirs can be represented as trapezoidal weirs (given that both rectangular and triangular weirs are special cases of the trapezoidal weir).

The effective head on a weir can be defined as:

$$H_{e,w} = \max\left(\max\left(H_{juw}, H_{jdw}\right) - (z_w + z_{inv,juw} + (1-\omega)y_{max,w}), 0\right) \tag{4.1}$$

The flow through a trapezoidal weir is the sum of the flow through the rectangular and triangular sections:

$$Q_w = C_{wR} L_w H_{e,w}^{3/2} + C_{wT} s_w H_{e,w}^{5/2} \tag{4.2}$$

The flow at the next time step can thus be estimated as:

$$Q_w^{t+\Delta t} = \frac{C_{wR} L_w H^t_{e,w} + C_{wT} s_w (H^t_{e,w})^2}{|Q^t_w|} H^{t+\Delta t}_{e,w} \tag{4.3}$$

Representing weirs in the solution matrix

Define the following indicator function:

$$\Omega(H_{juw}, H_{jdw}) = \begin{cases} 1, & H_{juw} \geqslant H_{jdw} \\ 0, & o/w \end{cases} \tag{4.4}$$

Similarly, define boolean-valued functions to represent the following flow conditions:

Submerged on high-head side

$$\Theta_{w,1} = \begin{cases} 1, & \Omega H_{juw} + (1-\Omega)H_{jdw} > z_w + z_{inv,juw} + (1-\omega)y_{max,w} \\ 0, & o/w \end{cases} \tag{4.5}$$

Submerged on low-head side

$$\Theta_{w,2} = \begin{cases} 1, & (1-\Omega)H_{juw} + \Omega H_{jdw} > z_o + z_{inv,juw} + (1-\omega)y_{max,w} \\ 0, & o/w \end{cases} \tag{4.6}$$

The flow through a weir can now be represented using the following linearized coefficient equation:

$$\boxed{Q_w^{t+\Delta t} = \alpha_w H_{juw}^{t+\Delta t} + \beta_w H_{jdw}^{t+\Delta t} + \chi_w} \tag{4.7}$$

Where:

$$\boxed{\alpha_w = \begin{cases} \gamma_w, & \Theta_{w,1} \wedge \Theta_{w,2} \\ \gamma_w \Omega (-1)^{1-\Omega}, & \Theta_{w,1} \wedge \neg\Theta_{w,2} \\ 0, & \neg\Theta_{w,1} \end{cases}} \tag{4.8}$$

$$\boxed{\beta_w = \begin{cases} -\gamma_w, & \Theta_{w,1} \wedge \Theta_{w,2} \\ \gamma_w (1-\Omega)(-1)^{1-\Omega}, & \Theta_{w,1} \wedge \neg\Theta_{w,2} \\ 0, & \neg\Theta_{w,1} \end{cases}} \tag{4.9}$$

$$\boxed{\chi_w = \begin{cases} 0, & \Theta_{w,1} \wedge \Theta_{w,2} \\ \gamma_w(-1)^{1-\Omega}[-z_{inv,juw} - z_w - (1-\omega)y_{max,w}], & \Theta_{w,1} \wedge \neg\Theta_{w,2} \\ 0, & \neg\Theta_{w,3} \end{cases}} \tag{4.10}$$

$$\boxed{\gamma_w = \frac{C_{wR}L_w H^t_{e,w} + C_{wT}s_w(H^t_{e,w})^2}{|Q^t_w|}} \tag{4.11}$$

Thus, define the matrix $A_W(t)$ and the vector $D_W(t)$. The elements of $A_W(t)$ are defined such that $A_{W_{j,j}}(t) = F^w_{j,j}$; $A_{W_{j,juw_\ell}} = \Phi^w_{j,juw_\ell}$; and $A_{W_{j,jdw_m}} = \Psi^w_{j,jdw_m}$ (with all other elements of $A_W(t)$ equal to zero).

$$\boxed{F^w_{j,j} = \sum_{m=1}^{NWUj} \alpha_{w_m} - \sum_{\ell=1}^{NWDj} \beta_{w_\ell}} \tag{4.12}$$

$$\boxed{\Phi^w_{j,juw_\ell} = -\alpha_{w_\ell}} \tag{4.13}$$

$$\boxed{\Psi^w_{j,jdw_m} = \beta_{w_m}} \tag{4.14}$$

Similarly, the elements of $D_W(t)$ are defined such that $D_{W_j} = G^w_j$.

$$\boxed{G^w_j = \sum_{m=1}^{NWUj} \chi_{w_m} - \sum_{\ell=1}^{NWDj} \chi_{w_\ell}} \tag{4.15}$$

S4.11 Representing pumps

Governing equations for pumps

The relationship between flow and head in a pump is usually defined by a pump curve. For this implementation, we assume that the flow/head relationship can be approximated by an ellipse centered at the origin defined over the support $[H_{min,p}, H_{max,p}]$.

First, define the effective head for the pump as follows:

$$H_{e,p} = \begin{cases} H_{max,p}, & H_{jdp} - H_{jup} > H_{max,p} \\ H_{jdp} - H_{jup}, & H_{min,p} < H_{jdp} - H_{jup} < H_{max,p} \\ H_{min,p}, & H_{jdp} - H_{jup} < H_{min,p} \end{cases} \tag{4.1}$$

Then, using the elliptical approximation, the flow through the pump can be represented as:

$$Q_p = \omega \sqrt{a_q^2 (1 - \frac{H_{e,p}^2}{a_h^2})} \tag{4.2}$$

Representing pumps in the solution matrix

Define boolean-valued functions to represent the following flow conditions:

Submerged inlet

$$\Theta_{p,1} = \begin{cases} 1, & H_{jup} \geqslant z_{inv,jup} + z_p \\ 0, & o/w \end{cases} \tag{4.3}$$

Head in pump curve range

$$\Theta_{p,2} = \begin{cases} 1, & H_{min,p} < H_{jdp} - H_{jup} < H_{max,p} \\ 0, & o/w \end{cases} \tag{4.4}$$

The flow through a pump can now be represented using the following linearized coefficient equation:

$$\boxed{Q_p^{t+\Delta t} = \alpha_p H_{jup}^{t+\Delta t} + \beta_p H_{jdp}^{t+\Delta t} + \chi_p} \tag{4.5}$$

$$\boxed{\alpha_p = \begin{cases} \gamma_p \omega^2, & \Theta_{p,1} \wedge \Theta_{p,2} \\ 0, & \Theta_{p,1} \wedge \neg\Theta_{p,2} \\ 0, & \neg\Theta_{p,1} \end{cases}} \tag{4.6}$$

$$\boxed{\beta_p = \begin{cases} -\gamma_p \omega^2, & \Theta_{p,1} \wedge \Theta_{p,2} \\ 0, & \Theta_{p,1} \wedge \neg\Theta_{p,2} \\ 0, & \neg\Theta_{p,1} \end{cases}} \tag{4.7}$$

$$\boxed{\chi_p = \begin{cases} \frac{a_q^2}{|Q_p^t|}, & \Theta_{p,1} \wedge \Theta_{p,2} \\ \omega\sqrt{a_q^2\left(1 - \frac{(H_{e,p}^t)^2}{a_h^2}\right)}, & \Theta_{p,1} \wedge \neg\Theta_{p,2} \\ 0, & \neg\Theta_{p,1} \end{cases}} \tag{4.8}$$

$$\boxed{\gamma_p = \frac{a_q^2 |H_{dp}^t - H_{up}^t|}{a_h^2 |Q_p^t|}} \tag{4.9}$$

Thus, define the matrix $A_P(t)$ and the vector $D_P(t)$. The elements of $A_P(t)$ are defined such that $A_{P_{j,j}}(t) = F_{j,j}^p$; $A_{P_{j,jup_\ell}} = \Phi_{j,jup_\ell}^p$; and $A_{P_{j,jdp_m}} = \Psi_{j,jdp_m}^p$ (with all other elements of $A_P(t)$ equal to zero).

$$\boxed{F_{j,j}^p = \sum_{m=1}^{NPUj} \alpha_{p_m} - \sum_{\ell=1}^{NPDj} \beta_{p_\ell}} \tag{4.10}$$

$$\boxed{\Phi_{j,jup_\ell}^p = -\alpha_{p_\ell}} \tag{4.11}$$

$$\boxed{\Psi_{j,jdp_m}^p = \beta_{p_m}} \tag{4.12}$$

Similarly, the elements of $D_P(t)$ are defined such that $D_{P_j} = G_j^p$.

$$\boxed{G_j^p = \sum_{m=1}^{NPUj} \chi_{p_m} - \sum_{\ell=1}^{NPDj} \chi_{p_\ell}} \tag{4.13}$$

Reformulation as an LTV system with controls

Combining together the governing equations for superlinks and dynamic control structures, we reformulate the hydraulic model's governing equations into the standard form of an implicit linear time-varying state space model:

$$A_1(t)\mathbf{x}(t+\Delta t) = A_2(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + D(t) \tag{4.14}$$

The left-hand side state transition matrix, A_1, is the sum of the corresponding left-hand solution matrices for superlinks, orifices, weirs and pumps.

$$A_1(t) = A_K(t) + A_O(t) + A_W(t) + A_P(t) + A_S(t) \tag{4.15}$$

The right-hand state transition matrix is a diagonal matrix that represents the current storage capacity of each superjunction, and is simply equal to A_s.

$$A_2(t) = A_S(t) \tag{4.16}$$

The constant term $D(t)$ is a vector representing the sum of the χ coefficients for superlinks, orifices, weirs and pumps.

$$D(t) = D_K(t) + D_O(t) + D_W(t) + D_P(t) \tag{4.17}$$

Note that for superjunctions with a fixed boundary head $H_{bc,j}^{t+\Delta t}$, the elements of the LTV system take the following values: $A_{1_{i,j}} = \{1 \; \forall \; i = j; \; 0 \; \forall \; i \neq j\}$, $A_{2_{j,j}} = 0 \; \forall \; j$, $\mathbf{u}(t)_j = 0$, and $D(t)_j = H_{bc,j}^{t+\Delta t}$.

S4.12 Mobile computational elements

Another major methodological advance that we introduce to the hydraulic solver is the idea of movable computational elements that track discontinuities in the water surface profile. Crucially, this modification enables better handling of backwater effects imposed by storage basins (such as retention basins and dams). Because storage elements often have a dominating effect on the system response, this modification allows for more accurate representation of real-world stormwater networks.

The original SUPERLINK formulation struggles to capture backwater effects induced by large storage elements because it is unable to represent discontinuities in the water surface at locations where no computational elements exist. To illustrate this problem, consider a superlink (k) that connects two superjunctions (j_1 and j_2) as shown in Figure

S4.3a. Moreover, assume that the head at the downstream junction lies between the invert elevations of junctions 2 and 3, and that there is no flow entering from either the upstream or downstream superjunctions. In this case, solving the recurrence relations will correctly predict the depths $h_2 = 0$ and $h_3 = H_2 - z_{inv,3}$. However, because there are no computational elements between these two junctions, the water surface in between them will be implicitly represented as a gradually-varied profile with slope $S_{o,2} + \frac{h_2 - h_3}{\Delta x_2}$. Because this water surface is not parallel with the horizontal, it will induce a flow Q_2 that will cause mass to accumulate in the downstream superjunction until the head at the downstream junction is equal to $H_2 = h_2 + z_{inv,2}$. This issue introduces mass balance errors into the model in situations where backwater profiles are present.

To handle backwater effects, we introduce movable computational elements that track discontinuities in the water surface profile. This scheme is illustrated in Figure S4.3b. At each time step, the movable computational element is repositioned within the superlink such that the invert elevation is level with the downstream superjunction head. When the internal recurrence relations are solved to determine the depths and flows within the superlink, the discontinuity in the water surface will be properly captured at the location of the movable element. Note that while our primary intention is to more accurately model backwater effects, this same scheme can be used to track other discontinuities in the water surface profile, such as locations of transcritical flow.

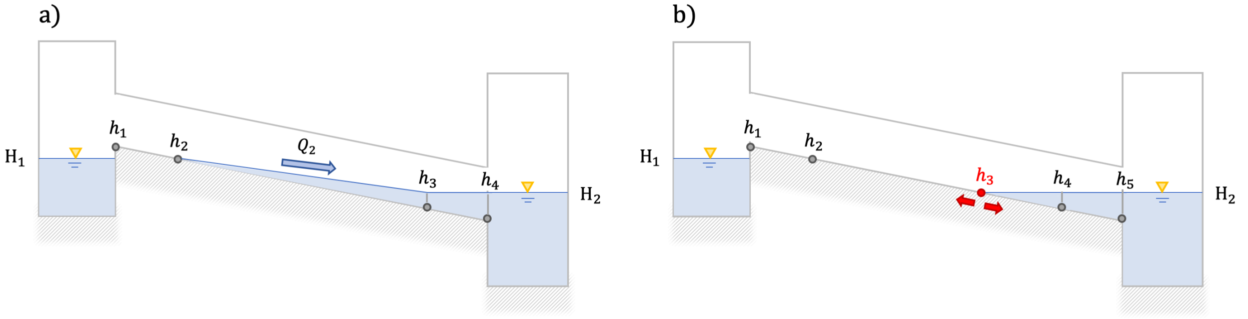


Figure S4.3. Mobile computational elements. Left: original superlink formulation. Right: superlink with mobile internal computational elements

S4.13 Step size control

Adaptive step size control is achieved using the digital filter approach proposed by Söderlind (2003) [3], using the generic stepsize recursion:

$$\Delta t_{n+1} = \left(\frac{\epsilon}{\hat{r}_n}\right)^{\zeta_1} \left(\frac{\epsilon}{\hat{r}_{n-1}}\right)^{\zeta_2} \left(\frac{\epsilon}{\hat{r}_{n-2}}\right)^{\zeta_3} \left(\frac{\Delta t_n}{\Delta t_{n-1}}\right)^{-\xi_2} \left(\frac{\Delta t_{n-1}}{\Delta t_{n-2}}\right)^{-\xi_3} \Delta t_n \tag{4.1}$$

Where Δt_{n+1} is the next time step size (at step $n+1$), Δt_n is the current time step size (at step n), $\hat{r}_n$ is the truncation error estimate at the current time step, and the ζ and ξ exponents are filter parameters. The variable ϵ is a user-specified fraction of the allowable tolerance:

$$\epsilon = \rho \cdot \mathrm{TOL} \tag{4.2}$$

Where $\rho < 1$ is some suitable safety factor. The variable $\hat{r}_n$ is an estimate of the truncation error at iteration n. Consider a generic differential function f. The discrete approximation of this function at time step $n+1$, $\hat{f}_{n+1}$, is given by the Backward Euler method as:

$$\hat{f}_{n+1} = \hat{f}_n + \Delta t \cdot f(\hat{f}_{n+1}, t_{n+1}) + \mathcal{O}(\Delta t)^2 \tag{4.3}$$

An estimate of the truncation error can be given by comparing the output at two different time steps.

Coarse step:

$$\hat{f}^{(0)}_{n+1} = \hat{f}_n + \Delta t \cdot f(\hat{f}_{n+1}, t_{n+1}) \tag{4.4}$$

Fine step:

$$\hat{f}_{n+\frac{1}{2}} = \hat{f}_n + \frac{\Delta t}{2} \cdot f(\hat{f}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) \tag{4.5}$$

$$\hat{f}^{(1)}_{n+1} = \hat{f}_{n+\frac{1}{2}} + \frac{\Delta t}{2} \cdot f(\hat{f}_{n+1}, t_{n+1}) \tag{4.6}$$

Then the truncation error estimate is given by:

$$\hat{r}_{n+1} = \hat{f}^{(1)}_{n+1} - \hat{f}^{(0)}_{n+1} \tag{4.7}$$

Table S4.5. Adaptive step filter parameters. Recommended filter parameters from Söderlind (2003) [142]. Note that k refers to the order of the error, which in the case of the Backward Euler method is $k = 2$.

$k\zeta_1$	$k\zeta_2$	$k\zeta_3$	ξ_2	ξ_3	Class
1/2	1/2	0	1/2	0	$H_0$211
1/6	1/6	0	0	0	H211 PI
1/4	1/2	1/4	3/4	1/4	$H_0$312
1/18	1/9	1/18	0	0	H312 PID
5/4	1/2	-3/4	-1/4	-3/4	$H_0$321
1/3	1/18	-5/18	-5/6	-1/6	H321

The choice of ζ and ξ parameters determines the properties of the filter. Recommended filter parameters are shown in Table S4.5.

S4.14 Hydrology

To enable `pipedream` to be used as a standalone stormwater management model, we include a fully-featured hydrologic module that computes infiltration and runoff using the Green-Ampt method. At each time step, the integrated form of the Green-Ampt equation is solved to estimate the cumulative infiltration depth for a soil element (indexed by f).

$$h_f^{t+\Delta t} = K_s \Delta t + h_f^t + \psi_f \theta_d \Big(\log(h_f^{t+\Delta t} + \psi_f \theta_d) - \log(h_f^t + \psi_f \theta_d) \Big) \tag{4.1}$$

Where $h_f^{t+\Delta t}$ is the cumulative infiltration depth at time $t + \Delta t$ (m), K_s is the saturated hydraulic conductivity (m/s), ψ_f is the suction head of the wetting front (m), and θ_d is the soil moisture deficit (unitless). The infiltration rate, i_f (m/s) is then estimated as:

$$i_f^{t+\Delta t} = \frac{h_f^{t+\Delta t} - h_f^t}{\Delta t} \tag{4.2}$$

The runoff rate per unit area is equal to the precipitation rate minus the infiltration rate:

$$q_f^{t+\Delta t} = p_f^{t+\Delta t} - i_f^{t+\Delta t} \tag{4.3}$$

Where $q_f^{t+\Delta t}$ is the runoff rate per unit area (m/s), and p_f is the precipitation rate per unit area (m/s).

The process of soil recovery (by which soil gradually dries due to evaporation and drainage), is implemented using the empirical method of Huber et al. (2005) [116].

To facilitate more realistic representation of overland flow, the infiltration model can be coupled with the hydraulic model. In this case (i) the runoff computed by the infiltration model is used as the lateral inflow to an overland flow element in the hydraulic model, and (ii) the routed depth from the hydraulic model is applied as the ponded depth in the infiltration calculation. By comparison, SWMM assumes that overland flow is instantaneously routed to the receiving channel, with a uniform flow rate given by Manning's equation. Coupling the infiltration and routing models enables a more realistic representation of the overland flow transport process, and also enables more accurate calculation of infiltration rates.

S4.15 Model comparisons

Scenario α

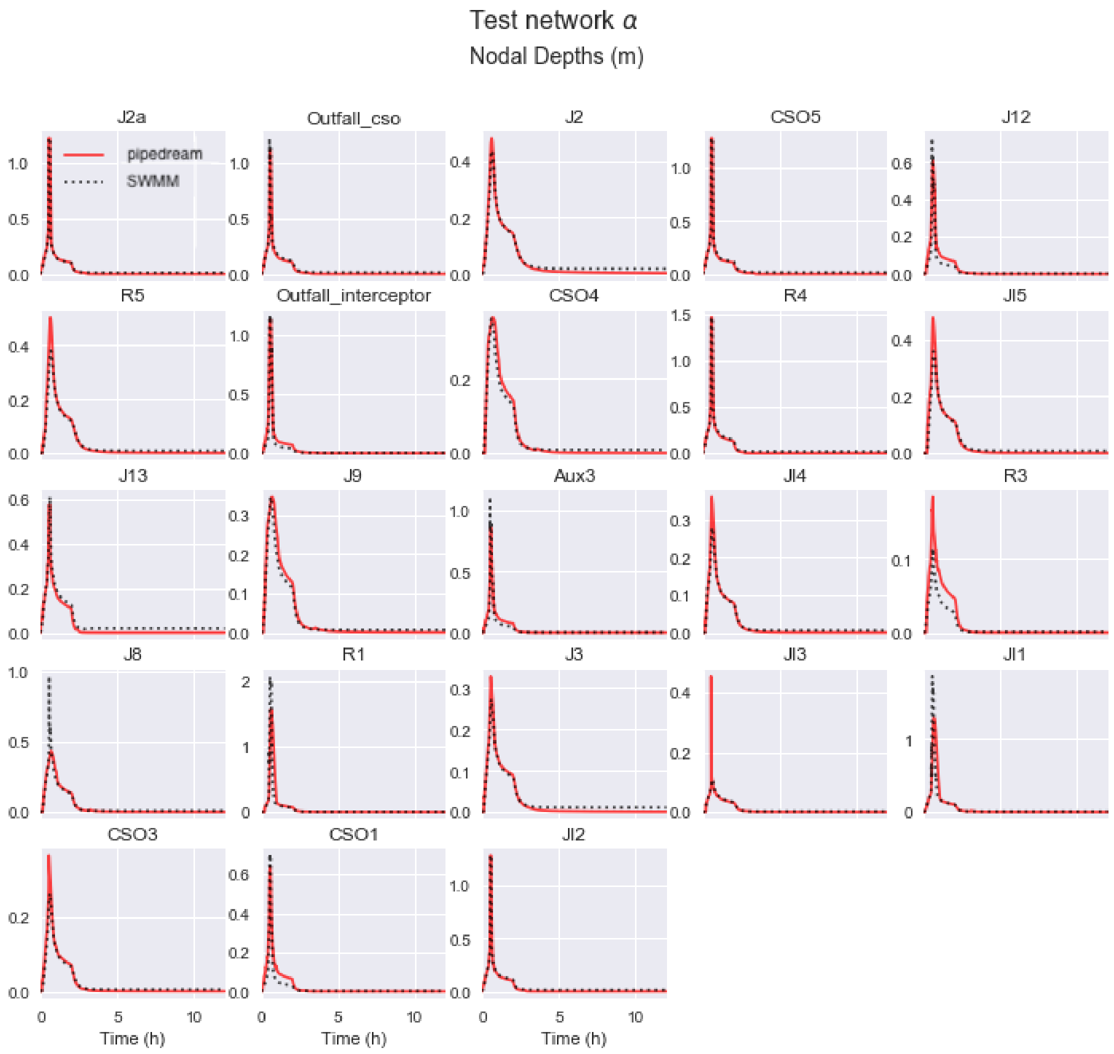


Figure S4.4. Model comparison: α. Model output for scenario alpha: comparison between pipedream and SWMM.

Scenario β

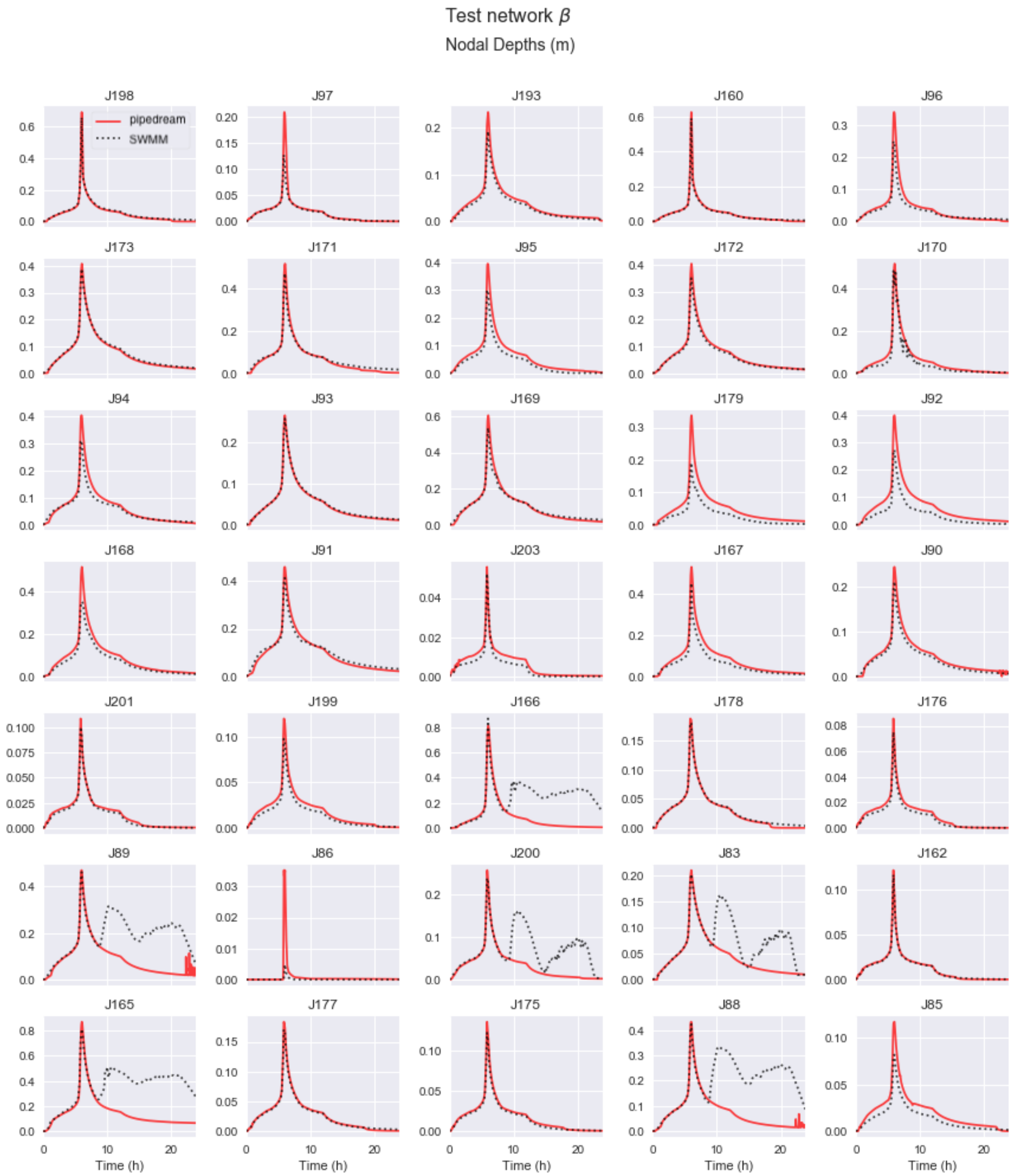


Figure S4.5. Model comparison: β**.** Model output for scenario beta: comparison between pipedream and SWMM (first 35 elements out of 210).

Scenario ϵ

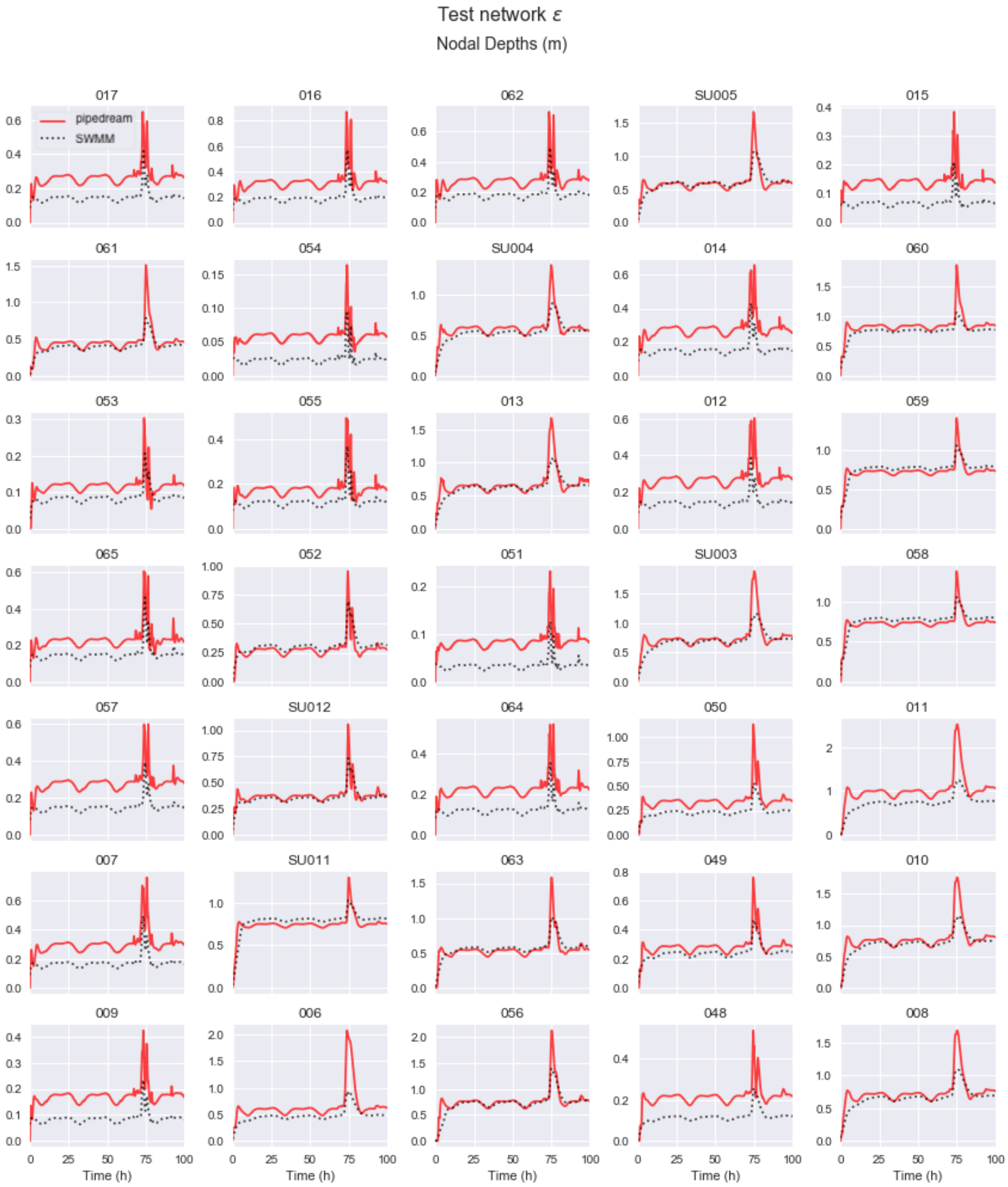


Figure S4.6. Model comparison: ϵ. Model output for scenario epsilon: comparison between pipedream and SWMM (first 35 elements out of 78).

S4.16 Kalman filtering as an alternative to rating curves

In addition to assisting with real-time operations, Kalman filtering may also prove useful for design and planning of water infrastructure by providing a robust method for estimating discharge from depth measurements. While water depth is significantly easier to measure, discharge measurements are indispensable for applications such as water resources planning, design of control structures, and estimation of environmental impacts. Traditionally, water resources engineers have used site-specific rating curves to estimate discharge based on stream stage observations. Rating curves are typically calibrated based on a limited number of flow measurements, and embed several assumptions about the underlying hydraulic system—namely, that there is a one-to-one relationship between depth and flow at a given site. In general, this assumption only holds under relatively special conditions (e.g. normal or critical flow), and may not be accurate for urban stormwater systems featuring control structures, water surface profile transitions, and highly unsteady flow through conduits (which may potentially be pressurized).

Through the application of Kalman filtering, discharge rates may be computed from measured depths using the full one-dimensional Saint-Venant equations, yielding a result that is both data-driven and theoretically justified. Figure S4.7 shows the discharge rates (left) and total discharged volume (right) for each site, as computed by the Kalman filter. In this case, the Kalman filter assimilates depth data from each site, and the resulting dynamical system is solved to compute discharges based on these updated depths. Compared to rating curves, Kalman filtering offers the added benefit of ensuring internal consistency between computed flows. Because the Kalman filter acts on the entire dynamical system at once, computed flows necessarily satisfy the mass and momentum balances that define the evolution of the system through time. This result is not guaranteed for rating curves, which may generate internally inconsistent flow estimates.

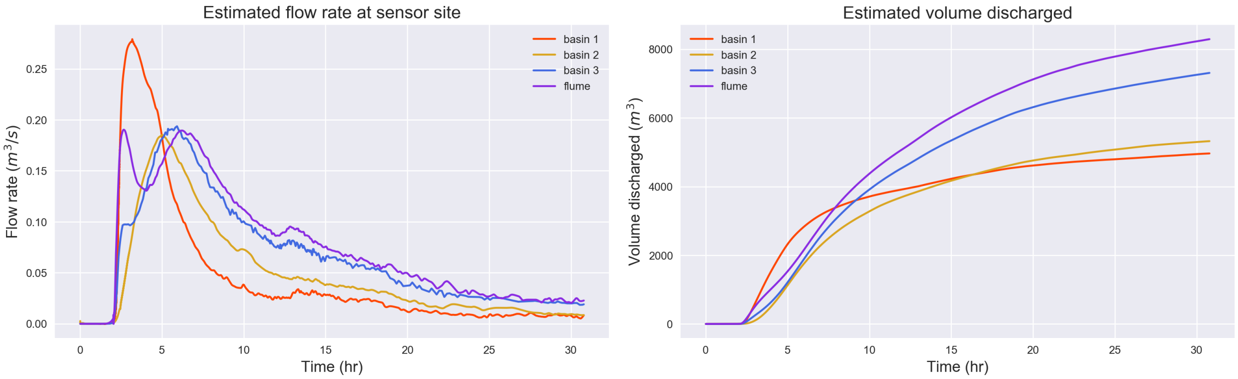


Figure S4.7. Estimation of flow using Kalman Filtering.

Chapter 5

A New Numerical Scheme for Water Quality Routing in Urban Drainage Networks

Abstract

Some of the most serious challenges posed by urban flooding pertain to its impacts on water quality; however, existing methods for modeling contaminant transport are often not well-suited for the complex and heterogeneous structure of stormwater networks. This chapter presents preliminary work towards a new numerical scheme for modeling contaminant fate and transport in urban drainage networks using the one-dimensional advection-reaction-diffusion equation. This numerical scheme can be paired with the hydrodynamic model developed in the previous chapter to enable detailed simulations of urban water quality. The numerical scheme is tested against analytical solutions to the advection-reaction-diffusion equation, and is found to be accurate and stable over large timesteps (on the order of several minutes). This numerical scheme will provide a vital tool for practitioners seeking to implement real-time modeling, state estimation and control of water quality in urban drainage systems.

5.1 Introduction

In addition to its direct impacts on human life and property, urban flooding presents major challenges for urban water quality. Intense flows associated with flooding often carry sediments, metals, nutrients and other contaminants into receiving water bodies, resulting in fish kills and algal blooms that decimate aquatic habitats [14–17]. For cities that rely on combined sewer systems, flooding can result in combined sewer overflows that discharge human and industrial waste into rivers and estuaries [150]. These incidents cost cities millions of dollars in fines and can negatively impact public health [150]. To plan for these contingencies, stormwater managers typically use contaminant transport models to estimate water quality impacts under future storm scenarios and evaluate how infrastructure expansions may mitigate these impacts.

The most frequently-used model for simulating water quality in urban drainage systems is the EPA Stormwater Management Model (SWMM) [116]. Despite its widespread use in industry, SWMM uses a relatively simplified model of contaminant transport. This formulation assumes that each element in the model is a continuously-stirred tank reactor in which contaminants are mixed instantaneously [43]. Contaminants are routed through the network by applying a mass balance to each element at each timestep [43]. While fast and intuitive, this formulation is difficult to reconcile with the physics of real-world drainage networks. In particular, this model is not well-suited to representing long channels in which contaminant transport is dominated by advection, and may give rise to misleading results.

The dynamics of contaminant transport in stormwater networks are more accurately described by the advection-diffusion equation—a hyperbolic partial differential equation that describes the evolution of a contaminant concentration in time and space when subjected to the motion of a carrier fluid [41]. When combined with relations describing reaction kinetics, this equation becomes the advection-reaction-diffusion (ARD) equation [41]. A large body of literature has been dedicated to solving the ARD equation in various contexts using an array of numerical techniques [42]—including explicit methods like Lax-Freidrichs, Lax-Wendroff, and Leapfrog schemes, or implicit methods like the Backward Euler and Crank-Nicolson schemes. However, stormwater networks present many particular challenges that are not handled well by canonical numerical schemes for the ARD equation. In particular, general-purpose numerical schemes typically assume that the problem domain is defined on a regular grid [42]. Stormwater networks, however, are composed of highly heterogenous elements including pressurized pipes, open channel sections, and large storage elements. While methods have been developed for solving the ARD equation in irregular geometries or meshless settings [151–153], these techniques are not necessarily amenable to networks composed of different types of computational elements that are governed by different dynamics. Methods for solving the ARD equation in stormwater networks must account for this heterogeneous structure by supplying suitable boundary relations at the interface between different elements.

In this chapter, I present preliminary work towards a new numerical scheme for simulating contaminant fate and transport in urban drainage networks. This scheme models contaminant dynamics by solving the one-dimensional ARD equation within stormwater networks using an implicit staggered-grid formulation. The numerical scheme is inspired by the SUPERLINK algorithm for solving the one-dimensional Saint-Venant equations in urban drainage networks [40]. In addition to its computational efficiency, this numerical scheme offers a number of advantages that make it especially suitable for real-time simulations:

- The numerical scheme is fully implicit and unconditionally stable[6]. This stability is important for online simulations in which the model is run continuously alongside a real-world system.

- The numerical scheme produces a state-space representation of the system that allows for the application of techniques from control theory—such as Kalman filtering for fusing sensor data into the model, or feedback control for regulating the concentration of contaminants within the system.

- The model structure is compatible with the online hydraulic model developed in the previous chapter, meaning that the two models can be combined for end-to-end real-time simulations of water quantity and quality.

[6] Note that stability is predicated on well-defined flow conditions. A no-flow condition can result in a rank-deficient or poorly-conditioned solution matrix which will lead to instability.

The following sections present the derivation of the numerical scheme, evaluate its implementation through comparison to analytical solutions of the ARD equation, and discuss the broader implications of this new methodology.

5.2 Methods

Theory

The transport of a contaminant through a control volume like a stormwater pipe is described by the one-dimensional advection-reaction-diffusion equation [41]:

$$\frac{\partial c}{\partial t} + \frac{\partial (uc)}{\partial x} - \frac{\partial}{\partial x}\left(D\frac{\partial c}{\partial x}\right) - r(c) = 0 \tag{5.1}$$

Where c is the concentration of the contaminant, u is the velocity of flow, D is the diffusion coefficient, $r(c)$ is the endogenous reaction rate, t is time and x is distance.

To solve this equation, one must supply a set of boundary conditions. For a stormwater system consisting of junctions connected by conduits, the boundary conditions for a conduit are given by the concentrations at the upstream and downstream junctions. Applying the continuity equation to the mass of the contaminant, the concentration at these junctions is given by the sum of inflows minus outflows minus any loss of material due to reactions:

$$\frac{d(Vc)}{dt} = Q_u c_u - Q_d c_d + Q_o c_o - Vr(c) \tag{5.2}$$

Where c is the concentration of the contaminant in the junction, V is the volume of solution in the junction, Q_u and Q_d are the upstream and downstream volumetric flow rates, c_u and c_d are the upstream and downstream concentrations, Q_o is the exogenous lateral inflow, c_o is the concentration of the exogenous lateral inflow, and $r(c)$ is the endogenous reaction rate.

To solve these equations in the general case, a numerical integration scheme is needed. The following sections derive a numerical scheme for solving these equations that is specifically adapted to the dynamics of stormwater networks.

Model structure and solution procedure

Before discussing the numerical scheme in detail, we first discuss the basic model structure. Stormwater systems typically consist of storage elements connected together by conduits. In storage elements, contaminant transport is dominated by mixing, while in conduits contaminant transport is dominated by advection. To account for this structure, the water quality model presented in this chapter uses four different types of computational elements inspired by the SUPERLINK algorithm for solving the one-dimensional Saint Venant equations for unsteady open-channel flow [40]. The four computational elements include superjunctions, superlinks, junctions, and links. Superjunctions are basic finite volumes that represent storage structures such as manholes or retention basins. Superlinks represent pipe or open channel sections and connect superjunctions together. Each superlink is segmented into a linear chain of n links and $n + 1$ junctions.

Within the staggered-grid numerical scheme proposed in this chapter, the advection-reaction-diffusion equation is applied to each link, while a mass balance equation is applied to each junction. Using these two sets of equations, recurrence relations are derived to relate the contaminant concentrations within each superlink to the contaminant concentrations at each superjunction. The system is then reformulated as a sparse matrix equation in which all unknown concentrations are expressed in terms of the concentrations at the superjunctions. After simultaneously solving for all unknown superjunction concentrations, the recurrence relations are used to solve for the internal concentrations within each superlink. The following sections describe the numerical scheme in detail.

Discretization of advection-reaction-diffusion equation

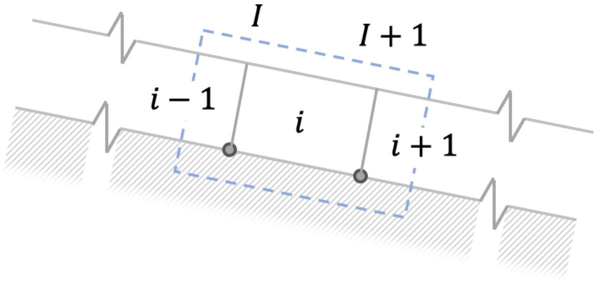


Figure 5.1. Control volume for advection-reaction-diffusion equation. Indexing scheme is shown centered around link i.

The advection-reaction-diffusion equation is discretized using a Backward Euler scheme, and applied to each link ik (see Figure 5.1 for an illustration of the link control volume):

$$\begin{aligned}\frac{1}{\Delta t}(\bar{c}_{ik}^{t+\Delta t} - \bar{c}_{ik}^{t}) + \frac{1}{\Delta x_{ik}}\left(u_{I+1k}c_{I+1k}^{t+\Delta t} - u_{Ik}c_{Ik}^{t+\Delta t}\right) \\ -D_{ik}\left(\frac{c_{I+1k}^{t+\Delta t} - \bar{c}_{ik}^{t+\Delta t}}{\frac{1}{2}\Delta x_{ik}^2} - \frac{\bar{c}_{ik}^{t+\Delta t} - c_{Ik}^{t+\Delta t}}{\frac{1}{2}\Delta x_{ik}^2}\right) - K_{ik}\bar{c}_{ik}^{t+\Delta t} = 0\end{aligned} \tag{5.3}$$

Where $\bar{c}_{ik}$ is the concentration in the link, c_{Ik} and c_{I+1k} are the concentrations at the upstream and downstream junctions, u_{Ik} and u_{I+1k} are the flow velocities at the upstream and downstream junctions, D_{ik} is the diffusion coefficient in the conduit, Δx_{ik} is the length of the conduit, and Δt is the time step. Here, it is assumed that the reaction rate can be represented by a first-order reaction with constant K_1:

$$r(c) = K_{ik}c \tag{5.4}$$

Combining terms, this equation can be written in terms of the following coefficient equation:

$$\alpha_{ik}c_{Ik}^{t+\Delta t} + \beta_{ik}\bar{c}_{ik}^{t+\Delta t} + \chi_{ik}c_{I+1k}^{t+\Delta t} = \gamma_{ik} \tag{5.5}$$

Where the coefficients are given by the following equations:

$$\alpha_{ik} = \frac{\hat{u}_{Ik}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2} \tag{5.6}$$

$$\beta_{ik} = \frac{1}{\Delta t} - \frac{\hat{u}_{Ik}}{\Delta x_{ik}} - \frac{\hat{u}_{I+1k}}{\Delta x_{ik}} + \frac{4D_{ik}}{\Delta x_{ik}^2} - K_{ik} \tag{5.7}$$

$$\chi_{ik} = \frac{\hat{u}_{I+1k}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2} \tag{5.8}$$

$$\gamma_{ik} = \frac{1}{\Delta t}\bar{c}_{ik}^{t} \tag{5.9}$$

Velocities are computed using the upwind advection scheme:

$$\hat{u}_{Ik} = -\max(u_{Ik}, 0) \tag{5.10}$$

$$\hat{u}_{I+1k} = -\max(-u_{I+1k}, 0) \tag{5.11}$$

Discretization of mass conservation

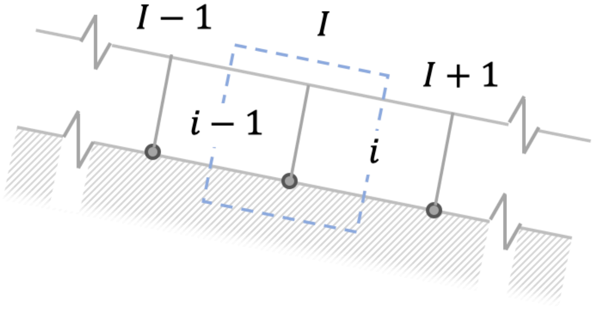


Figure 5.2. Control volume for mass conservation equation. Indexing scheme is shown centered around junction I.

Using a staggered-grid formulation, the mass conservation equation is applied around each junction Ik (see Figure 5.2 for an illustration of the junction control volume):

$$\begin{aligned}
\bar{c}_{ik}^{t+\Delta t} Q_{ik}^{t+\Delta t} - \bar{c}_{i-1k}^{t+\Delta t} Q_{i-1k}^{t+\Delta t} + \frac{A_{s,Ik}}{\Delta t}\left(c_{Ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} - c_{Ik}^{t} h_{Ik}^{t}\right) \\
+ \frac{B_{ik}\Delta x_{ik}}{2\Delta t}\left(\bar{c}_{ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} - \bar{c}_{ik}^{t} h_{Ik}^{t}\right) + \frac{B_{i-1k}\Delta x_{i-1k}}{2\Delta t}\left(\bar{c}_{i-1k}^{t+\Delta t} h_{Ik}^{t+\Delta t} - \bar{c}_{i-1k}^{t} h_{Ik}^{t}\right) \\
= c_{0,Ik}^{t+\Delta t} Q_{0,Ik}^{t+\Delta t} - \left(K_{Ik} A_{s,Ik} c_{Ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} + \frac{\Delta x_{ik}}{2} K_{ik} A_{ik} c_{ik}^{t+\Delta t} + \frac{\Delta x_{i-1k}}{2} K_{i-1k} A_{i-1k} c_{i-1k}^{t+\Delta t}\right)
\end{aligned} \tag{5.12}$$

Where Q_{ik} is the flow rate in link i, h_{Ik} is the depth in junction I, $A_{s,Ik}$ is the surface area of junction I, B_{ik} is the top width of flow in link i, $Q_{0,Ik}$ is the lateral overflow into the control volume, and $c_{0,Ik}$ is the contaminant concentration in the lateral overflow.

Through substitution, the discretized continuity equation can be represented in terms of the following coefficient equation:

$$\kappa_{Ik} \bar{c}_{i-1k}^{t+\Delta t} + \lambda_{Ik} c_{Ik}^{t+\Delta t} + \mu_{Ik} \bar{c}_{ik}^{t+\Delta t} = \eta_{Ik} \tag{5.13}$$

With the coefficients given by the following equations:

$$\kappa_{Ik} = -Q_{i-1k}^{t+\Delta t} + \frac{\Delta x_{i-1k}}{2}\left(\frac{B_{i-1k} h_{Ik}^{t+\Delta t}}{\Delta t} + K_{i-1k} A_{i-1k}\right) \tag{5.14}$$

$$\lambda_{Ik} = A_{s,Ik} h_{Ik}^{t+\Delta t} \left(\frac{1}{\Delta t} + K_{Ik} \right) \tag{5.15}$$

$$\mu_{Ik} = Q_{ik}^{t+\Delta t} + \frac{\Delta x_{ik}}{2} \left(\frac{B_{ik} h_{Ik}^{t+\Delta t}}{\Delta t} + K_{ik} A_{ik} \right) \tag{5.16}$$

$$\eta_{Ik} = c_{0,Ik}^{t+\Delta t} Q_{0,Ik}^{t+\Delta t} + \frac{A_{s,Ik} h_{Ik}^{t} c_{Ik}^{t}}{\Delta t} + \frac{B_{i-1k} \Delta x_{i-1k} h_{Ik}^{t} \bar{c}_{i-1k}^{t}}{2\Delta t} + \frac{B_{ik} \Delta x_{ik} h_{Ik}^{t} \bar{c}_{ik}^{t}}{2\Delta t} \tag{5.17}$$

Forming the solution matrix

The contaminant concentrations at each time step are determined by first solving for the contaminant concentrations within each superjunction, and then using the recurrence relations to solve for the internal contaminant concentrations within each superlink. Applying the mass balance equation to each superjunction, the change in contaminant mass within the superjunction is equal to the sum of inflows minus outflows and reaction losses:

$$\begin{aligned} \sum_{l=1}^{NKDj} \bar{c}_{dk_l}^{t+\Delta t} Q_{dk_l}^{t+\Delta t} - \sum_{m=1}^{NKUj} \bar{c}_{uk_m}^{t+\Delta t} Q_{uk_m}^{t+\Delta t} + \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} - K_j c_j^{t+\Delta t} V_j \\ = \frac{A_{sj}}{\Delta t} (c_j^{t+\Delta t} H_j^{t+\Delta t} - c_j^{t} H_j^{t}) \end{aligned} \tag{5.18}$$

Through a series of recurrence relations, it can be shown that the contaminant concentrations at the upstream and downstream ends of each superlink can be expressed as affine functions of the concentrations at the upstream and downstream superjunctions (see Section S5.5 in the Supplementary Information for details of the derivation):

$$\bar{c}_{uk}^{t+\Delta t} = \rho_{uk} c_{juk}^{t+\Delta t} + \tau_{uk} c_{jdk}^{t+\Delta t} + \omega_{uk} \tag{5.19}$$

$$\bar{c}_{dk}^{t+\Delta t} = \rho_{dk} c_{juk}^{t+\Delta t} + \tau_{dk} c_{jdk}^{t+\Delta t} + \omega_{dk} \tag{5.20}$$

Where $\bar{c}_{uk}$ and $\bar{c}_{dk}$ are the concentrations of contaminant at the upstream and downstream ends of superlink k, respectively; and c_{juk} and c_{jdk} are the concentrations of contaminant in the superjunctions upstream and downstream of superlink k, respectively.

The coefficients ρ_{uk}, τ_{uk}, ω_{uk}, ρ_{dk}, τ_{dk}, and ω_{dk} are relatively complex expressions that incorporate the continuity and advection-reaction-diffusion equations within superlink k.

Substituting the affine equations for the superlink boundary concentrations into the superjunction mass balance equation yields the following equation:

$$F^k_{j,j} c_j^{t+\Delta t} + \sum_{l=1}^{NKDj} \Phi^k_{j,juk_l} c_{juk_l}^{t+\Delta t} + \sum_{m=1}^{NKUj} \Psi^k_{j,jdk_m} c_{jdk_m}^{t+\Delta t} = \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} + \frac{A_{sj} H_j^t c_j^t}{\Delta t} + G_j^k \tag{5.21}$$

Where:

$$F^k_{j,j} = \frac{A_{sj} H_j^{t+\Delta t}}{\Delta t} + K_j V_j + \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t} \rho_{uk_m} - \sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} \tau_{dk_l} \tag{5.22}$$

$$\Phi^k_{j,juk_l} = -Q_{dk_l}^{t+\Delta t} \rho_{dk_l} \tag{5.23}$$

$$\Psi^k_{j,jdk_m} = Q_{uk_m}^{t+\Delta t} \tau_{uk_m} \tag{5.24}$$

$$G_j^k = \sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} \omega_{dk_l} - \sum_{l=1}^{NKUj} Q_{uk_m}^{t+\Delta t} \omega_{uk_m} \tag{5.25}$$

At each time step, this system of equations is solved to determine the contaminant concentrations at each superjunction. These concentrations are then substituted into the affine superlink boundary relations to determine the contaminant concentrations entering and exiting each superlink. These concentrations are in turn substituted into the recurrence relations to determine the concentrations within each internal junction and link.

Like the hydrodynamic model developed in the previous chapter, the water quality scheme is implemented in the Python programming language [144]. The `numba` just-in-time compiler is used to accelerate numerical code [145]. At each time step of the simulation, the water quality solver first imports hydraulic states from the hydrodynamic model (including flow rates, depths, cross-sectional areas of flow, and flow velocities). The water quality numerical scheme is then executed using the following steps:

1. Advection-reaction-diffusion coefficients α_{ik}, β_{ik},χ_{ik} and γ_{ik} are calculated for each link in the network (see Section S5.1).

2. Continuity coefficients κ_{Ik}, λ_{Ik},μ_{Ik} and η_{Ik} are calculated for each node in the network (see Section S5.2).

3. Forward recurrence coefficients U_{Ik}, V_{Ik} and W_{Ik} are computed (see Section S5.3).

4. Backward recurrence coefficients X_{Ik}, Y_{Ik} and Z_{Ik} are computed (see Section S5.4).

5. Superlink boundary coefficients ρ_{uk}, ρ_{dk}, τ_{uk}, τ_{dk}, ω_{uk} and ω_{dk} are computed (see Section S5.5).

6. The solution matrix is constructed using the superlink boundary coefficients, and is solved to determine the concentrations at each superjunction for the next time step, $c_j^{t+\Delta t}$ (see section S5.6).

7. Using the superjunction concentrations as the boundary conditions for each superlink, the recurrence relations are used to solve for the internal concentrations within each superlink.

5.3 Results

The numerical scheme is evaluated by comparing simulation results against two analytical solutions to the advection-reaction-diffusion equation. The first test investigates the case of a constant contaminant concentration applied to the upstream boundary of an open channel with uniform flow (i.e. the step response of the system). The second test case investigates the case of a point source of contamination applied as the initial condition (i.e. the impulse response of the system). Testing against analytical solutions ensures that the theoretical basis of the numerical scheme is sound, and also helps to ensure that the solver is free of serious implementation bugs.

Test 1: Constant concentration at upstream boundary

For the first test, we evaluate the ability of the numerical scheme to model contaminant transport in an open channel with an initial concentration of zero within the channel and

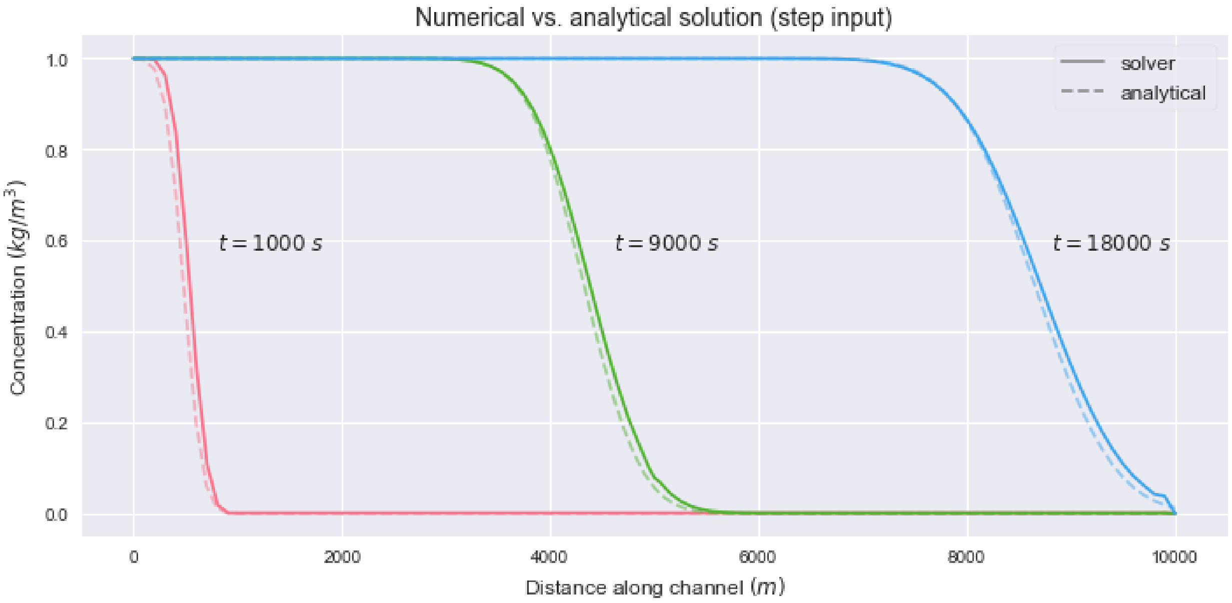


Figure 5.3. Step response test of water quality solver. Analytical vs. numerical solution, with a constant concentration of 1 $\mathrm{kg/m^3}$ applied to the upstream boundary.

a constant boundary concentration applied to the upstream end of the channel. Stated mathematically:

$$c(x, 0) = 0 \tag{5.1}$$

$$c(0, t) = c_0 \tag{5.2}$$

Under these conditions, the analytical solution to the advection-reaction-diffusion equation is given by:

$$c(x, t) = \frac{c_0}{2}\left(1 - \mathrm{erf}\left(-\frac{x}{4Dt}\right)\right) \tag{5.3}$$

Flow is assumed to be uniform with constant depth, velocity and discharge described by the well-known Manning's equation:

$$Q = \frac{A^{5/3}\sqrt{S_0}}{nP^{2/3}} \tag{5.4}$$

The channel is taken to be rectangular with a length of 10,000 m, a width of 2 m, a bottom slope of 0.0001 h:v, and a Manning's roughness of 0.01. The depth of flow is set to $h = 0.5$ m. Under these conditions, the flow is equal to $Q = 0.481\ \mathrm{m^3/s}$ and the velocity is equal to $u = 0.481\ m/s$. Applying these hydraulic states to each computational element in the channel, the numerical scheme is solved using a spatial discretization of 100 junctions and a time step of $t = 100\ s$.

Figure 5.3 shows the results of the constant boundary concentration test case. The numerical results show a strong correspondence with the analytical solution, with the numerical solution exhibiting a slight downwind bias. At a timestep of 100 s, the effects of numerical dispersion are limited—however, dispersion becomes more significant with decreasing time step sizes.

Test 2: Instantaneous point source concentration

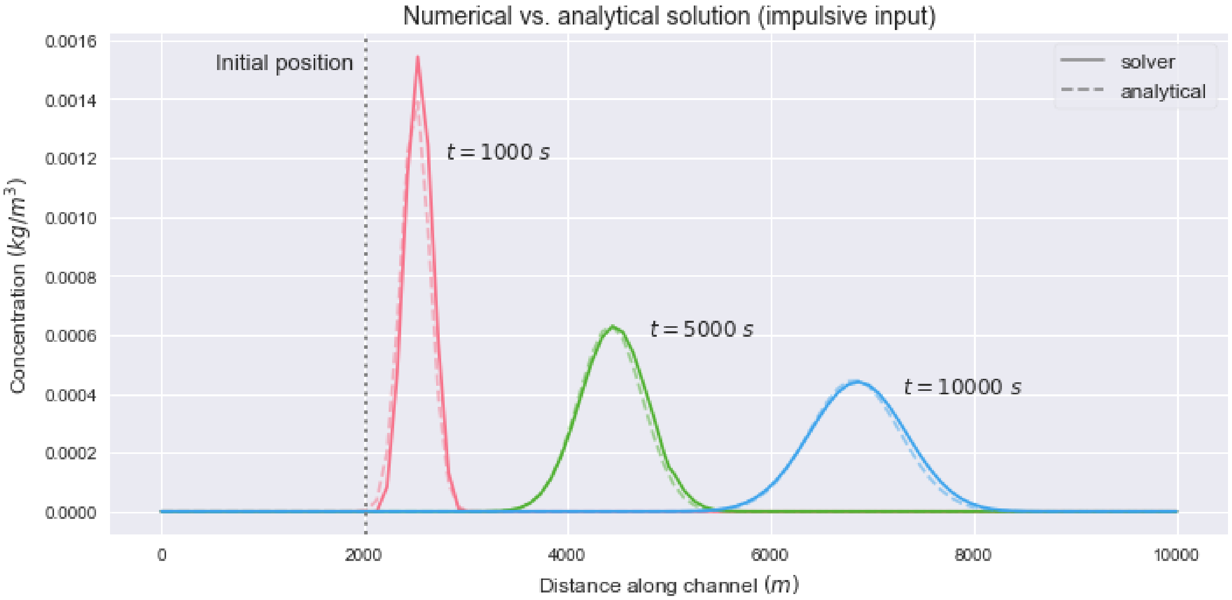


Figure 5.4. Impulse response test of water quality solver. Analytical vs. numerical solution, assuming an initial concentration of 10 kg/m^3 at $x = 2000\ m$.

For the second test, we evaluate the ability of the numerical scheme to model contaminant transport in an open channel with a point source of contaminant placed at an initial location x_0 within the channel and a concentration of zero everywhere else. Stated mathematically:

$$c(x, 0) = c_0\delta(x_0) \tag{5.5}$$

Under these conditions, the analytical solution to the advection-reaction-diffusion equation is given by:

$$c(x,t) = \frac{c_0}{A_0\sqrt{4\pi Dt}} \exp\left(-\frac{(x-(x_0+ut)^2}{4Dt}\right) \tag{5.6}$$

Where A_0 is the cross-sectional area of flow at initial location x_0. As with the previous test, flow is assumed to be uniform with the same flow rate, velocity, and hydraulic geometry. The same spatial and temporal discretization are also used.

Figure 5.4 shows the results of the instantaneous point source test case. As with the previous test case, the numerical solution shows a very slight downwind bias. However, the overall agreement between the numerical and analytical solutions is strong.

5.4 Discussion

The preliminary model presented in this chapter will provide system operators with the tools needed to run continuous real-time simulations of contaminant fate and transport in stormwater networks. The model is particularly well-suited to real-time simulations for a number of reasons. First, the numerical scheme is highly stable, which helps ensure that the model does not crash during continuous execution. Second, the model is capable of maintaining accuracy at large time steps. The ability to take large time steps is crucial because it ensures that the model is able to keep pace with real-world dynamics in large stormwater systems. Finally, much like the hydraulic model presented in the previous chapter, the water quality scheme admits a state space representation of system dynamics. This state space representation allows for the application of Kalman Filtering to fuse sensor data into the dynamical model, ensuring that system states are in continuous agreement with real-world dynamics.

In addition to providing the tools for continuous modeling, the numerical scheme presented in this chapter also provides a basis for real-time water quality control. Within the literature, a large number of studies have investigated the use of real-time control to improve water quality in urban drainage systems [20, 154]. However, due to a lack of interactive models with programmatic access, these studies have largely had to rely on surrogate models to implement control strategies. By exposing a state-space representation of system dynamics, the water quality model proposed in this study will enable practitioners to regulate water quality using powerful control algorithms such as model-predictive control and linear quadratic regulation.

5.5 Conclusion

This chapter presents preliminary work towards a new numerical scheme for modeling contaminant fate and transport in urban drainage networks. This numerical scheme uses a staggered-grid implicit discretization scheme inspired by the SUPERLINK hydrodynamic solver that emphasizes model stability at large time steps. We evaluate the numerical scheme by comparing simulation results against analytical solutions of the advection-reaction-diffusion equation and find that the numerical scheme is accurate and stable at time steps on the order of several minutes. For this reason, the numerical scheme is particularly suited for online modeling scenarios in which the simulation is executed alongside the real-world system in real-time. When paired with state estimation and feedback control techniques, the numerical scheme presented in this chapter will provide stormwater managers with a powerful tool for online monitoring and control of contaminants in stormwater networks.

Supplementary Information

S5.1 Discretization of advection/diffusion equation

Starting with the 1d advection-diffusion equation:

$$\frac{\partial c}{\partial t} + \frac{\partial (uc)}{\partial x} - \frac{\partial}{\partial x}\left(D\frac{\partial c}{\partial x}\right) - r(c) = 0 \tag{5.1}$$

Where c is the concentration of the contaminant, u is the velocity of flow, D is the diffusion coefficient, $r(c)$ is the endogenous reaction rate, t is time and x is distance.

For this application, we will assume that the reaction rate can be represented by a first-order reaction with constant K_1:

$$r(c) = K_{ik}c \tag{5.2}$$

The following discretization scheme can be applied to link i (see Fig. 5.1 for an illustration of the link control volume):

$$\begin{gathered} \frac{1}{\Delta t}(\bar{c}_{ik}^{t+\Delta t} - \bar{c}_{ik}^{t}) + \frac{1}{\Delta x_{ik}}\left(u_{I+1k}c_{I+1k}^{t+\Delta t} - u_{Ik}c_{Ik}^{t+\Delta t}\right) \\ -D_{ik}\left(\frac{c_{I+1k}^{t+\Delta t} - \bar{c}_{ik}^{t+\Delta t}}{\frac{1}{2}\Delta x_{ik}^2} - \frac{\bar{c}_{ik}^{t+\Delta t} - c_{Ik}^{t+\Delta t}}{\frac{1}{2}\Delta x_{ik}^2}\right) - K_{ik}\bar{c}_{ik}^{t+\Delta t} = 0 \end{gathered} \tag{5.3}$$

Where $\bar{c}_{ik}$ is the concentration in the link, c_{Ik} and c_{I+1k} are the concentrations at the upstream and downstream junctions, u_{Ik} and u_{I+1k} are the flow velocities at the upstream and downstream junctions, D_{ik} is the diffusion coefficient in the conduit, Δx_{ik} is the length of the conduit, and Δt is the time step.

Rearranging:

$$\left(-\frac{u_{Ik}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2}\right)c_{Ik}^{t+\Delta t} + \left(\frac{1}{\Delta t} + \frac{4D_{ik}}{\Delta x_{ik}^2} - K_{ik}\right)\bar{c}_{ik}^{t+\Delta t} + \left(\frac{u_{I+1k}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2}\right)c_{I+1k}^{t+\Delta t} = \frac{1}{\Delta t}\bar{c}_{ik}^{t} \tag{5.4}$$

This equation can be written in terms of the following coefficient equation:

$$\boxed{\alpha_{ik}c_{Ik}^{t+\Delta t} + \beta_{ik}\bar{c}_{ik}^{t+\Delta t} + \chi_{ik}c_{I+1k}^{t+\Delta t} = \gamma_{ik}} \tag{5.5}$$

Where:

$$\boxed{\alpha_{ik} = \frac{\hat{u}_{Ik}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2}} \tag{5.6}$$

$$\boxed{\beta_{ik} = \frac{1}{\Delta t} - \frac{\hat{u}_{Ik}}{\Delta x_{ik}} - \frac{\hat{u}_{I+1k}}{\Delta x_{ik}} + \frac{4D_{ik}}{\Delta x_{ik}^2} - K_{ik}} \tag{5.7}$$

$$\boxed{\chi_{ik} = \frac{\hat{u}_{I+1k}}{\Delta x_{ik}} - \frac{2D_{ik}}{\Delta x_{ik}^2}} \tag{5.8}$$

$$\boxed{\gamma_{ik} = \frac{1}{\Delta t}\bar{c}_{ik}^{t}} \tag{5.9}$$

And using the upwind advection scheme:

$$\boxed{\hat{u}_{Ik} = -\max(u_{Ik}, 0)} \tag{5.10}$$

$$\boxed{\hat{u}_{I+1k} = -\max(-u_{I+1k}, 0)} \tag{5.11}$$

S5.2 Discretization of mass conservation

The following discretization scheme can be applied to junction I (see Fig. 5.2 for an illustration of the junction control volume):

$$\begin{aligned}
&\bar{c}_{ik}^{t+\Delta t} Q_{ik}^{t+\Delta t} - \bar{c}_{i-1k}^{t+\Delta t} Q_{i-1k}^{t+\Delta t} + \frac{A_{s,Ik}}{\Delta t}(c_{Ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} - c_{Ik}^{t} h_{Ik}^{t}) \\
&+ \frac{B_{ik}\Delta x_{ik}}{2\Delta t}(\bar{c}_{ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} - \bar{c}_{ik}^{t} h_{Ik}^{t}) + \frac{B_{i-1k}\Delta x_{i-1k}}{2\Delta t}(\bar{c}_{i-1k}^{t+\Delta t} h_{Ik}^{t+\Delta t} - \bar{c}_{i-1k}^{t} h_{Ik}^{t}) \\
&= c_{0,Ik}^{t+\Delta t} Q_{0,Ik}^{t+\Delta t} - (K_{Ik} A_{s,Ik} c_{Ik}^{t+\Delta t} h_{Ik}^{t+\Delta t} + \frac{\Delta x_{ik}}{2} K_{ik} A_{ik} c_{ik}^{t+\Delta t} + \frac{\Delta x_{i-1k}}{2} K_{i-1k} A_{i-1k} c_{i-1k}^{t+\Delta t})
\end{aligned} \tag{5.1}$$

Where Q_{ik} is the flow rate in link i, h_{Ik} is the depth in junction I, $A_{s,Ik}$ is the surface area of junction I, B_{ik} is the top width of flow in link i, $Q_{0,Ik}$ is the lateral overflow into the control volume, and $c_{0,Ik}$ is the contaminant concentration in the lateral overflow.

Through substitution, the discretized continuity equation can be represented as follows:

$$\boxed{\kappa_{Ik}\bar{c}_{i-1k}^{t+\Delta t} + \lambda_{Ik}c_{Ik}^{t+\Delta t} + \mu_{Ik}\bar{c}_{ik}^{t+\Delta t} = \eta_{Ik}} \tag{5.2}$$

Where:

$$\boxed{\kappa_{Ik} = -Q_{i-1k}^{t+\Delta t} + \frac{\Delta x_{i-1k}}{2}\left(\frac{B_{i-1k}h_{Ik}^{t+\Delta t}}{\Delta t} + K_{i-1k}A_{i-1k}\right)} \tag{5.3}$$

$$\boxed{\lambda_{Ik} = A_{s,Ik}h_{Ik}^{t+\Delta t}\left(\frac{1}{\Delta t} + K_{Ik}\right)} \tag{5.4}$$

$$\boxed{\mu_{Ik} = Q_{ik}^{t+\Delta t} + \frac{\Delta x_{ik}}{2}\left(\frac{B_{ik}h_{Ik}^{t+\Delta t}}{\Delta t} + K_{ik}A_{ik}\right)} \tag{5.5}$$

$$\boxed{\eta_{Ik} = c_{0,Ik}^{t+\Delta t}Q_{0,Ik}^{t+\Delta t} + \frac{A_{s,Ik}h_{Ik}^{t}c_{Ik}^{t}}{\Delta t} + \frac{B_{i-1k}\Delta x_{i-1k}h_{Ik}^{t}\bar{c}_{i-1k}^{t}}{2\Delta t} + \frac{B_{ik}\Delta x_{ik}h_{Ik}^{t}\bar{c}_{ik}^{t}}{2\Delta t}} \tag{5.6}$$

S5.3 Forward recurrence

Starting with the advection-diffusion equation at the first link:

$$\alpha_{1k}c_{1k}^{t+\Delta t} + \beta_{1k}\bar{c}_{1k}^{t+\Delta t} + \chi_{1k}c_{2k}^{t+\Delta t} = \gamma_{1k} \tag{5.1}$$

Rewriting:

$$\boxed{\bar{c}_{1k}^{t+\Delta t} = U_{1k}c_{2k}^{t+\Delta t} + V_{1k} + W_{1k}c_{1k}^{t+\Delta t}} \tag{5.2}$$

Where:

$$\boxed{U_{1k} = -\frac{\chi_{1k}}{\beta_{1k}}} \tag{5.3}$$

$$\boxed{V_{1k} = \frac{\gamma_{1k}}{\beta_{1k}}} \tag{5.4}$$

$$\boxed{W_{1k} = -\frac{\alpha_{1k}}{\beta_{1k}}} \tag{5.5}$$

Next, advancing to the next element downstream:

$$\kappa_{2k}\bar{c}_{1k}^{t+\Delta t} + \lambda_{2k}c_{2k}^{t+\Delta t} + \mu_{2k}\bar{c}_{2k}^{t+\Delta t} = \eta_{2k} \tag{5.6}$$

$$\alpha_{2k}c_{2k}^{t+\Delta t} + \beta_{2k}\bar{c}_{2k}^{t+\Delta t} + \chi_{2k}c_{3k}^{t+\Delta t} = \gamma_{2k} \tag{5.7}$$

Substituting $\bar{c}_{1k}$ into the continuity equation:

$$\kappa_{2k}(U_{1k}c_{2k}^{t+\Delta t} + V_{1k} + W_{1k}c_{1k}^{t+\Delta t}) + \lambda_{2k}c_{2k}^{t+\Delta t} + \mu_{2k}\bar{c}_{2k}^{t+\Delta t} = \eta_{2k} \tag{5.8}$$

Isolating c_{2k}:

$$c_{2k}^{t+\Delta t} = \frac{\eta_{2k} - \kappa_{2k}V_{1k} - \kappa_{2k}W_{1k}c_{1k}^{t+\Delta t} - \mu_{2k}\bar{c}_{2k}^{t+\Delta t}}{\lambda_{2k} + \kappa_{2k}U_{1k}} \tag{5.9}$$

Substituting this expression back into the advection-diffusion equation:

$$\alpha_{2k}\left(\frac{\eta_{2k} - \kappa_{2k}V_{1k} - \kappa_{2k}W_{1k}c_{1k}^{t+\Delta t} - \mu_{2k}\bar{c}_{2k}^{t+\Delta t}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right) + \beta_{2k}\bar{c}_{2k}^{t+\Delta t} + \chi_{2k}c_{3k}^{t+\Delta t} = \gamma_{2k} \tag{5.10}$$

Collecting terms:

$$\begin{aligned}\left(-\frac{\alpha_{2k}\kappa_{2k}W_{1k}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right)c_{1k}^{t+\Delta t} + \left(\chi_{2k}\right)c_{3k}^{t+\Delta t} + \left(\beta_{2k} - \frac{\alpha_{2k}\mu_{2k}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right)\bar{c}_{2k}^{t+\Delta t} \\ + \left(\frac{\alpha_{2k}\eta_{2k} - \alpha_{2k}\kappa_{2k}V_{1k}}{\lambda_{2k} + \kappa_{2k}U_{1k}} - \gamma_{2k}\right) = 0\end{aligned} \tag{5.11}$$

Expressing in terms of $\bar{c}_{2k}$:

$$\begin{aligned}\left(\beta_{2k} - \frac{\alpha_{2k}\mu_{2k}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right)\bar{c}_{2k}^{t+\Delta t} = \left(\frac{\alpha_{2k}\kappa_{2k}W_{1k}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right)c_{1k}^{t+\Delta t} + \left(-\chi_{2k}\right)c_{3k}^{t+\Delta t} \\ + \left(\gamma_{2k} - \frac{\alpha_{2k}\eta_{2k} - \alpha_{2k}\kappa_{2k}V_{1k}}{\lambda_{2k} + \kappa_{2k}U_{1k}}\right)\end{aligned} \tag{5.12}$$

Thus:

$$\bar{c}_{2k}^{t+\Delta t} = U_{2k}c_{3k}^{t+\Delta t} + V_{2k} + W_{2k}c_{1k}^{t+\Delta t} \tag{5.13}$$

Where:

$$U_{2k} = -\frac{\chi_{2k}}{T_{2k}} \tag{5.14}$$

$$V_{2k} = \frac{\gamma_{2k}}{T_{2k}} - \frac{\alpha_{2k}\eta_{2k} - \alpha_{2k}\kappa_{2k}V_{1k}}{T_{2k}(\lambda_{2k} + \kappa_{2k}U_{1k})} \tag{5.15}$$

$$W_{2k} = \frac{\alpha_{2k}\kappa_{2k}W_{1k}}{T_{2k}(\lambda_{2k} + \kappa_{2k}U_{1k})} \tag{5.16}$$

$$T_{2k} = \beta_{2k} - \frac{\alpha_{2k}\mu_{2k}}{\lambda_{2k} + \kappa_{2k}U_{1k}} \tag{5.17}$$

Thus, we can define the following recurrence relation for the forward direction:

$$\boxed{\bar{c}_{ik}^{t+\Delta t} = U_{Ik}c_{I+1k}^{t+\Delta t} + V_{Ik} + W_{Ik}c_{1k}^{t+\Delta t}} \tag{5.18}$$

Where:

$$\boxed{U_{Ik} = -\frac{\chi_{ik}}{T_{ik}}} \tag{5.19}$$

$$\boxed{V_{Ik} = \frac{\gamma_{Ik}}{T_{ik}} - \frac{\alpha_{ik}(\eta_{Ik} - \kappa_{ik}V_{I-1k})}{T_{ik}(\lambda_{Ik} + \kappa_{Ik}U_{I-1k})}} \tag{5.20}$$

$$\boxed{W_{Ik} = \frac{\alpha_{ik}\kappa_{Ik}W_{I-1k}}{T_{ik}(\lambda_{Ik} + \kappa_{Ik}U_{I-1k})}} \tag{5.21}$$

$$\boxed{T_{ik} = \beta_{ik} - \frac{\alpha_{ik}\mu_{Ik}}{\lambda_{Ik} + \kappa_{Ik}U_{I-1k}}} \tag{5.22}$$

S5.4 Backward recurrence

Starting with the advection-diffusion equation at the final link:

$$\alpha_{nk}c_{Nk}^{t+\Delta t} + \beta_{nk}\bar{c}_{nk}^{t+\Delta t} + \chi_{nk}c_{N+1k}^{t+\Delta t} = \gamma_{nk} \tag{5.1}$$

Rewriting:

$$\boxed{\bar{c}_{nk}^{t+\Delta t} = X_{Nk}c_{Nk}^{t+\Delta t} + Y_{Nk} + Z_{Nk}c_{N+1k}^{t+\Delta t}} \tag{5.2}$$

Where:

$$\boxed{X_{Nk} = -\frac{\alpha_{nk}}{\beta_{nk}}} \tag{5.3}$$

$$\boxed{Y_{Nk} = \frac{\gamma_{nk}}{\beta_{nk}}} \tag{5.4}$$

$$\boxed{Z_{Nk} = -\frac{\chi_{nk}}{\beta_{nk}}} \tag{5.5}$$

Advancing to the next element upstream:

$$\kappa_{Nk}\bar{c}^{t+\Delta t}_{n-1k} + \lambda_{Nk}c^{t+\Delta t}_{Nk} + \mu_{Nk}\bar{c}^{t+\Delta t}_{nk} = \eta_{Nk} \tag{5.6}$$

$$\alpha_{n-1k}c^{t+\Delta t}_{N-1k} + \beta_{n-1k}\bar{c}^{t+\Delta t}_{n-1k} + \chi_{n-1k}c^{t+\Delta t}_{Nk} = \gamma_{n-1k} \tag{5.7}$$

Substituting $\bar{c}_{nk}$ into the continuity equation:

$$\kappa_{Nk}\bar{c}^{t+\Delta t}_{n-1k} + \lambda_{Nk}c^{t+\Delta t}_{Nk} + \mu_{Nk}(X_{Nk}c^{t+\Delta t}_{Nk} + Y_{Nk} + Z_{Nk}c^{t+\Delta t}_{N+1k}) = \eta_{Nk} \tag{5.8}$$

Isolating c_{Nk}:

$$c^{t+\Delta t}_{Nk} = \frac{\eta_{Nk} - \mu_{Nk}Y_{Nk} - \mu_{Nk}Z_{Nk}c^{t+\Delta t}_{N+1k} - \kappa_{Nk}\bar{c}^{t+\Delta t}_{n-1k}}{\lambda_{Nk} + \mu_{Nk}X_{Nk}} \tag{5.9}$$

Substituting this expression back into the advection-diffusion equation:

$$\alpha_{n-1k}c^{t+\Delta t}_{N-1k} + \beta_{n-1k}\bar{c}^{t+\Delta t}_{n-1k} + \chi_{n-1k}\left(\frac{\eta_{Nk} - \mu_{Nk}Y_{Nk} - \mu_{Nk}Z_{Nk}c^{t+\Delta t}_{N+1k} - \kappa_{Nk}\bar{c}^{t+\Delta t}_{n-1k}}{\lambda_{Nk} + \mu_{Nk}X_{Nk}}\right) = \gamma_{n-1k} \tag{5.10}$$

Collecting terms:

$$\begin{aligned} \Big(\alpha_{n-1k}\Big)c^{t+\Delta t}_{N-1k} + \left(\beta_{n-1k} - \frac{\chi_{n-1k}\kappa_{Nk}}{\lambda_{Nk} + \mu_{Nk}X_{Nk}}\right)\bar{c}^{t+\Delta t}_{n-1k} \\ + \left(\frac{-\chi_{n-1k}\mu_{Nk}Z_{Nk}}{\lambda_{Nk} + \mu_{Nk}X_{Nk}}\right)c^{t+\Delta t}_{N+1k} + \left(\frac{\chi_{n-1k}\eta_{Nk} - \chi_{n-1k}\mu_{Nk}Y_{Nk}}{\lambda_{Nk} + \mu_{Nk}X_{Nk}} - \gamma_{n-1k}\right) = 0 \end{aligned} \tag{5.11}$$

Expressing in terms of $\bar{c}_{n-1k}$:

$$\begin{aligned}\left(\beta_{n-1k}-\frac{\chi_{n-1k}\kappa_{Nk}}{\lambda_{Nk}+\mu_{Nk}X_{Nk}}\right)\bar{c}_{n-1k}^{t+\Delta t}=\left(-\alpha_{n-1k}\right)c_{N-1k}^{t+\Delta t}+\left(\frac{\chi_{n-1k}\mu_{Nk}Z_{Nk}}{\lambda_{Nk}+\mu_{Nk}X_{Nk}}\right)c_{N+1k}^{t+\Delta t}\\ +\left(\gamma_{n-1k}-\frac{\chi_{n-1k}\eta_{Nk}-\chi_{n-1k}\mu_{Nk}Y_{Nk}}{\lambda_{Nk}+\mu_{Nk}X_{Nk}}\right)\end{aligned}\tag{5.12}$$

Thus:

$$\bar{c}_{n-1k}^{t+\Delta t}=X_{N-1k}c_{N-1k}^{t+\Delta t}+Y_{N-1k}+Z_{N-1k}c_{N+1k}^{t+\Delta t}\tag{5.13}$$

Where:

$$X_{N-1k}=\frac{-\alpha_{n-1k}}{O_{n-1k}}\tag{5.14}$$

$$Y_{N-1k}=\frac{\gamma_{n-1k}}{O_{n-1k}}-\frac{\chi_{n-1k}\eta_{Nk}-\chi_{n-1k}\mu_{Nk}Y_{Nk}}{O_{n-1k}(\lambda_{Nk}+\mu_{Nk}X_{Nk})}\tag{5.15}$$

$$Z_{N-1k}=\frac{\chi_{n-1k}\mu_{Nk}Z_{Nk}}{O_{n-1k}(\lambda_{Nk}+\mu_{Nk}X_{Nk})}\tag{5.16}$$

$$O_{n-1k}=\beta_{n-1k}-\frac{\chi_{n-1k}\kappa_{Nk}}{\lambda_{Nk}+\mu_{Nk}X_{Nk}}\tag{5.17}$$

Thus, we can define the following recurrence relation for the backwards direction:

$$\boxed{\bar{c}_{ik}^{t+\Delta t}=X_{Ik}c_{Ik}^{t+\Delta t}+Y_{Ik}+Z_{Ik}c_{N+1k}^{t+\Delta t}}\tag{5.18}$$

Where:

$$\boxed{X_{Ik}=\frac{-\alpha_{ik}}{O_{ik}}}\tag{5.19}$$

$$\boxed{Y_{Ik}=\frac{\gamma_{ik}}{O_{ik}}-\frac{\chi_{ik}(\eta_{I+1k}-\mu_{I+1k}Y_{I+1k})}{O_{ik}(\lambda_{I+1k}+\mu_{I+1k}X_{I+1k})}}\tag{5.20}$$

$$\boxed{Z_{Ik}=\frac{\chi_{ik}\mu_{I+1k}Z_{I+1k}}{O_{ik}(\lambda_{I+1k}+\mu_{I+1k}X_{I+1k})}}\tag{5.21}$$

$$\boxed{O_{ik}=\beta_{ik}-\frac{\chi_{ik}\kappa_{I+1k}}{\lambda_{I+1k}+\mu_{I+1k}X_{I+1k}}}\tag{5.22}$$

S5.5 Setting system boundary conditions

From the recurrence relations:

$$\bar{c}_{nk}^{t+\Delta t} = U_{Nk} c_{N+1k}^{t+\Delta t} + V_{Nk} + W_{Nk} c_{1k}^{t+\Delta t} \tag{5.1}$$

$$\bar{c}_{1k}^{t+\Delta t} = X_{1k} c_{1k}^{t+\Delta t} + Y_{1k} + Z_{1k} c_{N+1k}^{t+\Delta t} \tag{5.2}$$

Recall the continuity equations for the boundary junctions:

$$\kappa_{1k} \bar{c}_{uk}^{t+\Delta t} + \lambda_{1k} c_{1k}^{t+\Delta t} + \mu_{1k} \bar{c}_{1k}^{t+\Delta t} = \eta_{1k} \tag{5.3}$$

$$\kappa_{N+1k} \bar{c}_{nk}^{t+\Delta t} + \lambda_{N+1k} c_{N+1k}^{t+\Delta t} + \mu_{N+1k} \bar{c}_{dk}^{t+\Delta t} = \eta_{N+1k} \tag{5.4}$$

Substituting into the boundary recurrence relations:

$$\frac{1}{\kappa_{N+1k}}(\eta_{N+1k} - \lambda_{N+1k} c_{N+1k}^{t+\Delta t} - \mu_{N+1k} \bar{c}_{dk}^{t+\Delta t}) = U_{Nk} c_{N+1k}^{t+\Delta t} + V_{Nk} + W_{Nk} c_{1k}^{t+\Delta t} \tag{5.5}$$

$$\bar{c}_{dk}^{t+\Delta t} = \frac{-\kappa_{N+1k} U_{Nk} - \lambda_{N+1k}}{\mu_{N+1k}} c_{N+1k}^{t+\Delta t} + \frac{\eta_{N+1k} - \kappa_{N+1k} V_{Nk}}{\mu_{N+1k}} + \frac{-\kappa_{N+1k} W_{Nk}}{\mu_{N+1k}} c_{1k}^{t+\Delta t} \tag{5.6}$$

$$\frac{1}{\mu_{1k}}(\eta_{1k} - \kappa_{1k} \bar{c}_{uk}^{t+\Delta t} - \lambda_{1k} c_{1k}^{t+\Delta t}) = X_{1k} c_{1k}^{t+\Delta t} + Y_{1k} + Z_{1k} c_{N+1k}^{t+\Delta t} \tag{5.7}$$

$$\bar{c}_{uk}^{t+\Delta t} = \frac{-\mu_{1k} X_{1k} - \lambda_{1k}}{\kappa_{1k}} c_{1k}^{t+\Delta t} + \frac{\eta_{1k} - \mu_{1k} Y_{1k}}{\kappa_{1k}} + \frac{-\mu_{1k} Z_{1k}}{\kappa_{1k}} c_{N+1k}^{t+\Delta t} \tag{5.8}$$

Thus, the concentrations at the inlet and outlet can be described as:

$$\bar{c}_{uk}^{t+\Delta t} = X_{uk} c_{1k}^{t+\Delta t} + Y_{uk} + Z_{uk} c_{N+1k}^{t+\Delta t} \tag{5.9}$$

$$\bar{c}_{dk}^{t+\Delta t} = U_{dk} c_{N+1k}^{t+\Delta t} + V_{dk} + W_{dk} c_{1k}^{t+\Delta t} \tag{5.10}$$

Where:

$$X_{uk} = \frac{-\mu_{1k} X_{1k} - \lambda_{1k}}{\kappa_{1k}} \tag{5.11}$$

$$Y_{uk} = \frac{\eta_{1k} - \mu_{1k} Y_{1k}}{\kappa_{1k}} \tag{5.12}$$

$$Z_{uk} = \frac{-\mu_{1k} Z_{1k}}{\kappa_{1k}} \tag{5.13}$$

$$U_{dk} = \frac{-\kappa_{N+1k} U_{Nk} - \lambda_{N+1k}}{\mu_{N+1k}} \tag{5.14}$$

$$V_{dk} = \frac{\eta_{N+1k} - \kappa_{N+1k} V_{Nk}}{\mu_{N+1k}} \tag{5.15}$$

$$W_{dk} = \frac{-\kappa_{N+1k} W_{Nk}}{\mu_{N+1k}} \tag{5.16}$$

To establish boundary conditions for each superlink, we apply the advection/diffusion relation to the boundary:

$$\alpha_{uk} c_{juk}^{t+\Delta t} + \beta_{uk} \bar{c}_{uk}^{t+\Delta t} + \chi_{uk} c_{1k}^{t+\Delta t} = \gamma_{uk} \tag{5.17}$$

$$\alpha_{dk} c_{N+1k}^{t+\Delta t} + \beta_{dk} \bar{c}_{dk}^{t+\Delta t} + \chi_{dk} c_{jdk}^{t+\Delta t} = \gamma_{dk} \tag{5.18}$$

Thus:

$$c_{1k} = \theta_{uk} \bar{c}_{uk} + \sigma_{uk} c_{juk} + \xi_{uk} \tag{5.19}$$

$$c_{N+1k} = \theta_{dk} \bar{c}_{dk} + \sigma_{dk} c_{jdk} + \xi_{dk} \tag{5.20}$$

Where:

$$\theta_{uk} = \frac{-\beta_{uk}}{\chi_{uk}} \tag{5.21}$$

$$\sigma_{uk} = \frac{-\alpha_{uk}}{\chi_{uk}} \tag{5.22}$$

$$\xi_{uk} = \frac{\gamma_{uk}}{\chi_{uk}} \tag{5.23}$$

$$\theta_{dk} = \frac{-\beta_{dk}}{\alpha_{dk}} \tag{5.24}$$

$$\sigma_{dk} = \frac{-\chi_{dk}}{\alpha_{dk}} \tag{5.25}$$

$$\xi_{dk} = \frac{\gamma_{dk}}{\alpha_{dk}} \tag{5.26}$$

Plugging this in to the previous relation:

$$\bar{c}_{uk}^{t+\Delta t} = X_{uk}(\theta_{uk}\bar{c}_{uk} + \sigma_{uk}c_{juk} + \xi_{uk}) + Y_{uk} + Z_{uk}(\theta_{dk}\bar{c}_{dk} + \sigma_{dk}c_{jdk} + \xi_{dk}) \tag{5.27}$$

$$\bar{c}_{dk}^{t+\Delta t} = U_{dk}(\theta_{dk}\bar{c}_{dk} + \sigma_{dk}c_{jdk} + \xi_{dk}) + V_{dk} + W_{dk}(\theta_{uk}\bar{c}_{uk} + \sigma_{uk}c_{juk} + \xi_{uk}) \tag{5.28}$$

Rearranging:

$$(1 - X_{uk}\theta_{uk})\bar{c}_{uk}^{t+\Delta t} + (-Z_{uk}\theta_{dk})\bar{c}_{dk} = (X_{uk}\sigma_{uk})c_{juk} + (Z_{uk}\sigma_{dk})c_{jdk} + \pi_{uk} \tag{5.29}$$

$$(-W_{dk}\theta_{uk})\bar{c}_{uk} + (1 - U_{dk}\theta_{dk})\bar{c}_{dk} = (W_{dk}\sigma_{uk})c_{juk} + (U_{dk}\sigma_{dk})c_{jdk} + \pi_{dk} \tag{5.30}$$

Where:

$$\pi_{uk} = Y_{uk} + X_{uk}\xi_{uk} + Z_{uk}\xi_{dk} \tag{5.31}$$

$$\pi_{dk} = V_{dk} + U_{dk}\xi_{dk} + W_{dk}\xi_{uk} \tag{5.32}$$

Writing as a matrix equation:

$$\begin{bmatrix} (1-X_{uk}\theta_{uk}) & -Z_{uk}\theta_{dk} \\ -W_{dk}\theta_{uk} & (1-U_{dk}\theta_{dk}) \end{bmatrix} \begin{bmatrix} \bar{c}_{uk} \\ \bar{c}_{dk} \end{bmatrix} = \begin{bmatrix} (X_{uk}\sigma_{uk})c_{juk} + (Z_{uk}\sigma_{dk})c_{jdk} + \pi_{uk} \\ (W_{dk}\sigma_{uk})c_{juk} + (U_{dk}\sigma_{dk})c_{jdk} + \pi_{dk} \end{bmatrix} \tag{5.33}$$

Computing the inverse:

$$\begin{bmatrix} \bar{c}_{uk} \\ \bar{c}_{dk} \end{bmatrix} = \frac{1}{D_k^*} \begin{bmatrix} (1-U_{dk}\theta_{dk}) & Z_{uk}\theta_{dk} \\ W_{dk}\theta_{uk} & (1-X_{uk}\theta_{uk}) \end{bmatrix} \begin{bmatrix} (X_{uk}\sigma_{uk})c_{juk} + (Z_{uk}\sigma_{dk})c_{jdk} + \pi_{uk} \\ (W_{dk}\sigma_{uk})c_{juk} + (U_{dk}\sigma_{dk})c_{jdk} + \pi_{dk} \end{bmatrix} \tag{5.34}$$

Where:

$$D_k^* = (1-X_{uk}\theta_{uk})(1-U_{dk}\theta_{dk}) - Z_{uk}W_{dk}\theta_{uk}\theta_{dk} \tag{5.35}$$

Carrying out the multiplication:

$$\begin{aligned} \bar{c}_{uk} = \big\{&(1-U_{dk}\theta_{dk})[(X_{uk}\sigma_{uk})c_{juk} + (Z_{uk}\sigma_{dk})c_{jdk} + \pi_{uk}] \\ &+Z_{uk}\theta_{dk}[(W_{dk}\sigma_{uk})c_{juk} + (U_{dk}\sigma_{dk})c_{jdk} + \pi_{dk}]\big\}/D_k^* \end{aligned} \tag{5.36}$$

$$\begin{aligned} \bar{c}_{dk} = \big\{&W_{dk}\theta_{uk}[(X_{uk}\sigma_{uk})c_{juk} + (Z_{uk}\sigma_{dk})c_{jdk} + \pi_{uk}] \\ &+(1-X_{uk}\theta_{uk})[(W_{dk}\sigma_{uk})c_{juk} + (U_{dk}\sigma_{dk})c_{jdk} + \pi_{dk}]\big\}/D_k^* \end{aligned} \tag{5.37}$$

Thus:

$$\begin{aligned} \bar{c}_{uk} = \big\{&[(1-U_{dk}\theta_{dk})X_{uk}\sigma_{uk} + Z_{uk}\theta_{dk}W_{dk}\sigma_{uk}]c_{juk} \\ &+[(1-U_{dk}\theta_{dk})Z_{uk}\sigma_{dk} + Z_{uk}\theta_{dk}U_{dk}\sigma_{dk}]c_{jdk} \\ &+[(1-U_{dk}\theta_{dk})\pi_{uk} + Z_{uk}\theta_{dk}\pi_{dk}]\big\}/D_k^* \end{aligned} \tag{5.38}$$

$$\begin{aligned} \bar{c}_{dk} = \big\{&[W_{dk}\theta_{uk}X_{uk}\sigma_{uk} + (1-X_{uk}\theta_{uk})W_{dk}\sigma_{uk}]c_{juk} \\ &+[W_{dk}\theta_{uk}Z_{uk}\sigma_{dk} + (1-X_{uk}\theta_{uk})U_{dk}\sigma_{dk}]c_{jdk} \\ &+[W_{dk}\theta_{uk}\pi_{uk} + (1-X_{uk}\theta_{uk})\pi_{dk}]\big\}/D_k^* \end{aligned} \tag{5.39}$$

Thus, the concentrations at the superlink boundaries can be described as a function of the upstream and downstream superjunction concentrations:

$$\boxed{\bar{c}_{uk}^{t+\Delta t} = \rho_{uk} c_{juk}^{t+\Delta t} + \tau_{uk} c_{jdk}^{t+\Delta t} + \omega_{uk}} \tag{5.40}$$

$$\boxed{\bar{c}_{dk}^{t+\Delta t} = \rho_{dk} c_{juk}^{t+\Delta t} + \tau_{dk} c_{jdk}^{t+\Delta t} + \omega_{dk}} \tag{5.41}$$

Where:

$$\boxed{\rho_{uk} = \frac{(1 - U_{dk}\theta_{dk})X_{uk}\sigma_{uk} + Z_{uk}\theta_{dk}W_{dk}\sigma_{uk}}{D_k^*}} \tag{5.42}$$

$$\boxed{\tau_{uk} = \frac{(1 - U_{dk}\theta_{dk})Z_{uk}\sigma_{dk} + Z_{uk}\theta_{dk}U_{dk}\sigma_{dk}}{D_k^*}} \tag{5.43}$$

$$\boxed{\omega_{uk} = \frac{(1 - U_{dk}\theta_{dk})(Y_{uk} + X_{uk}\xi_{uk} + Z_{uk}\xi_{dk}) + Z_{uk}\theta_{dk}(V_{dk} + U_{dk}\xi_{dk} + W_{dk}\xi_{uk})}{D_k^*}} \tag{5.44}$$

$$\boxed{\rho_{dk} = \frac{W_{dk}\theta_{uk}X_{uk}\sigma_{uk} + (1 - X_{uk}\theta_{uk})W_{dk}\sigma_{uk}}{D_k^*}} \tag{5.45}$$

$$\boxed{\tau_{dk} = \frac{W_{dk}\theta_{uk}Z_{uk}\sigma_{dk} + (1 - X_{uk}\theta_{uk})U_{dk}\sigma_{dk}}{D_k^*}} \tag{5.46}$$

$$\boxed{\omega_{dk} = \frac{W_{dk}\theta_{uk}(Y_{uk} + X_{uk}\xi_{uk} + Z_{uk}\xi_{dk}) + (1 - X_{uk}\theta_{uk})(V_{dk} + U_{dk}\xi_{dk} + W_{dk}\xi_{uk})}{D_k^*}} \tag{5.47}$$

$$\boxed{D_k^* = (1 - X_{uk}\theta_{uk})(1 - U_{dk}\theta_{dk}) - Z_{uk}W_{dk}\theta_{uk}\theta_{dk}} \tag{5.48}$$

S5.6 Forming the solution matrix

The mass balance of contaminants at a superjunction is given by:

$$\begin{aligned}\sum_{l=1}^{NKDj} \bar{c}_{dk_l}^{t+\Delta t} Q_{dk_l}^{t+\Delta t} - \sum_{m=1}^{NKUj} \bar{c}_{uk_m}^{t+\Delta t} Q_{uk_m}^{t+\Delta t} + \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} - K_j c_j^{t+\Delta t} V_j \\ = \frac{A_{sj}}{\Delta t}(c_j^{t+\Delta t} H_j^{t+\Delta t} - c_j^t H_j^t)\end{aligned} \tag{5.1}$$

Plugging in the expressions from the recurrence relations:

$$\begin{aligned}\sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t}(\rho_{dk_l} c_{juk_l}^{t+\Delta t} + \tau_{dk_l} c_{jdk_l}^{t+\Delta t} + \omega_{dk_l}) \\ - \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t}(\rho_{uk_m} c_{juk_m}^{t+\Delta t} + \tau_{uk_m} c_{jdk_m}^{t+\Delta t} + \omega_{uk_m}) \\ + \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} = \frac{A_{sj}}{\Delta t}(c_j^{t+\Delta t} H_j^{t+\Delta t} - c_j^t H_j^t) + K_j V_j c_j^{t+\Delta t}\end{aligned} \tag{5.2}$$

Because $c_{jdk_l} = c_j$ and $c_{juk_m} = c_j$:

$$\begin{aligned}\sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t}(\rho_{dk_l} c_{juk_l}^{t+\Delta t} + \tau_{dk_l} c_{j}^{t+\Delta t} + \omega_{dk_l}) \\ - \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t}(\rho_{uk_m} c_{j}^{t+\Delta t} + \tau_{uk_m} c_{jdk_m}^{t+\Delta t} + \omega_{uk_m}) \\ + \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} = \frac{A_{sj}}{\Delta t}(c_j^{t+\Delta t} H_j^{t+\Delta t} - c_j^t H_j^t) + K_j V_j c_j^{t+\Delta t}\end{aligned} \tag{5.3}$$

Rearranging:

$$\begin{aligned}\left(\frac{A_{sj} H_j^{t+\Delta t}}{\Delta t} + K_j V_j + \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t} \rho_{uk_m} - \sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} \tau_{dk_l}\right) c_j^{t+\Delta t} \\ + \sum_{m=1}^{NKUj} Q_{uk_m}^{t+\Delta t} \tau_{uk_m} c_{jdk_m}^{t+\Delta t} - \sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} \rho_{dk_l} c_{juk_l}^{t+\Delta t} \\ = \bar{c}_{o,j}^{t+\Delta t} Q_{o,j}^{t+\Delta t} + \frac{A_{sj} H_j^t}{\Delta t} + \sum_{l=1}^{NKDj} Q_{dk_l}^{t+\Delta t} \omega_{dk_l} - \sum_{l=1}^{NKUj} Q_{uk_m}^{t+\Delta t} \omega_{uk_m}\end{aligned} \tag{5.4}$$

Reducing to a coefficient equation:

$$\boxed{F^k_{j,j} c_j^{t+\Delta t} + \sum_{l=1}^{NKDj} \Phi^k_{j,juk_l} c^{t+\Delta t}_{juk_l} + \sum_{m=1}^{NKUj} \Psi^k_{j,jdk_m} c^{t+\Delta t}_{jdk_m} = \bar{c}^{t+\Delta t}_{o,j} Q^{t+\Delta t}_{o,j} + \frac{A_{sj} H^t_j c^t_j}{\Delta t} + G^k_j} \tag{5.5}$$

Where:

$$\boxed{F^k_{j,j} = \frac{A_{sj} H^{t+\Delta t}_j}{\Delta t} + K_j V_j + \sum_{m=1}^{NKUj} Q^{t+\Delta t}_{uk_m} \rho_{uk_m} - \sum_{l=1}^{NKDj} Q^{t+\Delta t}_{dk_l} \tau_{dk_l}} \tag{5.6}$$

$$\boxed{\Phi^k_{j,juk_l} = -Q^{t+\Delta t}_{dk_l} \rho_{dk_l}} \tag{5.7}$$

$$\boxed{\Psi^k_{j,jdk_m} = Q^{t+\Delta t}_{uk_m} \tau_{uk_m}} \tag{5.8}$$

$$\boxed{G^k_j = \sum_{l=1}^{NKDj} Q^{t+\Delta t}_{dk_l} \omega_{dk_l} - \sum_{l=1}^{NKUj} Q^{t+\Delta t}_{uk_m} \omega_{uk_m}} \tag{5.9}$$

Solution matrix equation for example network

For brevity, define $\tilde{G}_j = \frac{A_{sj}}{\Delta t} H^t_j c^t_j + c_{o,j} Q_{o,j} + G_j$. Thus, for the example network in Ji (1998), the sparse matrix equation at time $t + \Delta t$ is expressed as:

$$\begin{bmatrix} F^k_{1,1} & \Psi^k_{1,2} & 0 & 0 & 0 & 0 \\ \Phi^k_{2,1} & F^k_{2,2} & \Psi^k_{2,3} & 0 & \Psi^k_{2,5} & 0 \\ 0 & \Phi^k_{3,2} & F^k_{3,3} & \Psi^k_{3,4} & \Phi^k_{3,5} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \Phi^k_{5,2} & \Psi^k_{5,3} & 0 & F^k_{5,5} & \Psi^k_{5,6} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^{t+\Delta t} \\ c_2^{t+\Delta t} \\ c_3^{t+\Delta t} \\ c_4^{t+\Delta t} \\ c_5^{t+\Delta t} \\ c_6^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \tilde{G}^k_1 \\ \tilde{G}^k_2 \\ \tilde{G}^k_3 \\ c^{t+\Delta t}_{bc,4} \\ \tilde{G}^k_5 \\ c^{t+\Delta t}_{bc,6} \end{bmatrix} \tag{5.10}$$

To recover the superjunction concentrations, the matrix equation is solved for the unknown left-hand side vector at each time step.

Accounting for control structures

For orifices, weirs, and pumps let:

$$\bar{c}_{o}^{t+\Delta t} = \Omega_{o} c_{juo}^{t+\Delta t} + (1 - \Omega_{o}) c_{jdo}^{t+\Delta t} \tag{5.11}$$

$$\bar{c}_{w}^{t+\Delta t} = \Omega_{w} c_{juw}^{t+\Delta t} + (1 - \Omega_{w}) c_{jdw}^{t+\Delta t} \tag{5.12}$$

$$\bar{c}_{p}^{t+\Delta t} = \Omega_{p} c_{jup}^{t+\Delta t} + (1 - \Omega_{p}) c_{jdp}^{t+\Delta t} \tag{5.13}$$

Where:

$$\Omega_{o} = \begin{cases} 1, & Q_{o} \geqslant 0 \\ 0, & Q_{o} < 0 \end{cases} \tag{5.14}$$

$$\Omega_{w} = \begin{cases} 1, & Q_{w} \geqslant 0 \\ 0, & Q_{w} < 0 \end{cases} \tag{5.15}$$

$$\Omega_{p} = \begin{cases} 1, & Q_{p} \geqslant 0 \\ 0, & Q_{p} < 0 \end{cases} \tag{5.16}$$

To account for control structures like orifices, weirs, and pumps, we can modify the superjunction continuity equation.

$$\boxed{\begin{aligned} [F^k_{j,j}+F^o_{j,j}+F^w_{j,j}+F^o_{j,j}]c_j^{t+\Delta t}\\ +\sum_{l=1}^{NKDj}\Phi^k_{j,juk_l}c^{t+\Delta t}_{juk_l}+\sum_{m=1}^{NKUj}\Psi^k_{j,jdk_m}c^{t+\Delta t}_{jdk_m}\\ +\sum_{l=1}^{NODj}\Phi^o_{j,juo_l}c^{t+\Delta t}_{juo_l}+\sum_{m=1}^{NOUj}\Psi^o_{j,jdo_m}c^{t+\Delta t}_{jdo_m}\\ +\sum_{l=1}^{NWDj}\Phi^w_{j,juw_l}c^{t+\Delta t}_{juw_l}+\sum_{m=1}^{NWUj}\Psi^w_{j,jdw_m}c^{t+\Delta t}_{jdw_m}\\ +\sum_{l=1}^{NPDj}\Phi^p_{j,jup_l}c^{t+\Delta t}_{jup_l}+\sum_{m=1}^{NPUj}\Psi^p_{j,jdp_m}c^{t+\Delta t}_{jdp_m}\\ =\bar{c}^{t+\Delta t}_{o,j}Q^{t+\Delta t}_{o,j}+\frac{A_{sj}H_j^t c_j^t}{\Delta t}+G_j \end{aligned}} \tag{5.17}$$

Where:

Orifices

$$\boxed{F^o_{j,j}=\sum_{m=1}^{NOUj}Q^{t+\Delta t}_{o_m}\Omega_{o_m}-\sum_{l=1}^{NODj}Q^{t+\Delta t}_{o_l}(1-\Omega_{o_l})} \tag{5.18}$$

$$\boxed{\Phi_{j,juo_l}=-Q^{t+\Delta t}_{o_l}\Omega_{o_l}} \tag{5.19}$$

$$\boxed{\Psi_{j,jdo_m}=Q^{t+\Delta t}_{o_m}(1-\Omega_{o_m})} \tag{5.20}$$

Weirs

$$\boxed{F^w_{j,j}=\sum_{m=1}^{NWUj}Q^{t+\Delta t}_{w_m}\Omega_{w_m}-\sum_{l=1}^{NWDj}Q^{t+\Delta t}_{w_l}(1-\Omega_{w_l})} \tag{5.21}$$

$$\boxed{\Phi_{j,juw_l}=-Q^{t+\Delta t}_{w_l}\Omega_{w_l}} \tag{5.22}$$

$$\boxed{\Psi_{j,jdw_m}=Q^{t+\Delta t}_{w_m}(1-\Omega_{w_m})} \tag{5.23}$$

Pumps

$$\boxed{F^{p}_{j,j} = \sum_{m=1}^{NPUj} Q_{p_m}^{t+\Delta t}\Omega_{p_m} - \sum_{l=1}^{NPDj} Q_{p_l}^{t+\Delta t}(1-\Omega_{p_l})} \tag{5.24}$$

$$\boxed{\Phi_{j,jup_l} = -Q_{p_l}^{t+\Delta t}\Omega_{p_l}} \tag{5.25}$$

$$\boxed{\Psi_{j,jdp_m} = Q_{p_m}^{t+\Delta t}(1-\Omega_{p_m})} \tag{5.26}$$

Chapter 6

Hydrograph Peak-Shaving Using a Graph-Theoretic Algorithm for Placement of Hydraulic Control Structures

Abstract

The need to attenuate hydrograph peaks is central to the design of stormwater and flood control systems. However, few guidelines exist for siting hydraulic control structures such that system-scale benefits are maximized. This study presents a new graph-theoretic algorithm for stabilizing the hydrologic response of watersheds by placing controllers at strategic locations in the drainage network. This algorithm identifies subcatchments that dominate the peak of the hydrograph, and then finds the "cuts" in the drainage network that maximally isolate these subcatchments, thereby flattening the hydrologic response. Evaluating the performance of the algorithm through an ensemble of hydrodynamic simulations, we find that our controller placement algorithm produces consistently flatter discharges than randomized controller configurations—both in terms of the peak discharge and the overall variance of the hydrograph. By attenuating flashy flows, our algorithm provides a powerful methodology for mitigating flash floods, reducing erosion, and protecting aquatic ecosystems. More broadly, we show that controller placement exerts an important influence on the hydrologic response and demonstrate that analysis of drainage network structure can inform more effective stormwater control policies.

6.1 Introduction

In the wake of rapid urbanization, aging infrastructure and a changing climate, effective stormwater management poses a major challenge for cities worldwide [19]. Flash floods are one of the largest causes of natural disaster deaths in the developed world [1], and often occur when stormwater systems fail to convey runoff from urban areas [18]. At the same time, many cities suffer from impaired water quality due to inadequate stormwater control [14]. Flashy flows erode streambeds, release sediment-bound pollutants, and damage aquatic habitats [14–17], while untreated runoff may trigger fish kills and toxic algal blooms [155, 156]. Engineers have historically responded to these problems by expanding and upsizing stormwater control infrastructure [11]. However, larger infrastructure frequently brings adverse side-effects, such as dam-induced disruption of riparian ecosystems [157], and erosive discharges due to overdesigned conveyance infrastructure [19]. As a result, recent work has called for the replacement of traditional peak attenuation infrastructure with targeted solutions that better reduce environmental impacts [13, 158].

As the drawbacks of oversized stormwater infrastructure become more apparent, many cities are turning towards decentralized stormwater solutions to regulate and treat urban runoff while reducing adverse impacts. Green infrastructure, for instance, uses low-impact rain gardens, bioswales, and green roofs to condition flashy flows and remove contaminants [159–161]. *Smart* stormwater systems take this idea further by retrofitting static infrastructure with dynamically controlled valves, gates and pumps [19, 60, 102, 162]. By actuating small, distributed storage basins and conveyance structures in real-time, *smart* stormwater systems can halt combined sewer overflows [12], mitigate flooding [19], and improve water quality at a fraction of the cost of new construction [19, 102]. While decentralized stormwater management tools show promise towards mitigating urban water problems, it is currently unclear how these systems can be designed to achieve maximal benefits at the watershed scale. Indeed, some research suggests when stormwater control facilities are not designed in a global context, local best management practices can lead to adverse system-scale outcomes—in some cases inducing downstream flows that are more intense than those produced under unregulated conditions [57, 163].

Thus, as cities begin to experiment with decentralized stormwater control, the question of *where* to place control structures becomes crucial. While many studies have investigated the ways in which active control can realize system-scale benefits (using techniques like feedback control [44], market-based control [12], or model-predictive control, [20, 154]), the location of control structures within the drainage network may serve an equally important function. Hydrologists have long recognized the role that drainage network topology plays in shaping hydrologic response [164–172]. It follows that strategic placement of hydraulic control structures can shape the hydrograph to fulfill operational objectives, such as maximally flattening flood waves and regulating erosion downstream. To date, however, little research has been done to assess the problem of optimal placement of hydraulic control structures in drainage networks:

- Recent studies have investigated optimal placement of green infrastructure upgrades like green roofs, rain tanks and bioswales [46–48, 173–176]. However, these studies generally focus on quantifying the potential benefits of green infrastructure projects through representative case studies [47, 48, 173, 174], and do not intend to present a generalized framework for placement of stormwater control structures. As a result, many of these studies focus on optimizing multiple objectives (such as urban heat island mitigation [175], air quality [176], or quality of life considerations [46]), or use complex socio-physical models and optimization frameworks [48], making it difficult to draw general conclusions about controller placement in drainage networks.

- Studies of pressurized water distribution networks have investigated the related problems of valve placement [177, 178], sensor placement [179], subnetwork vulnerability assessment [180], and network sectorization [181, 182]. While these studies provide valuable insights into the ways that complex network theory can inform drinking water infrastructure design, water distribution networks are pressure-driven and cyclic, and are thus governed by different dynamics than natural drainage networks, which are mainly gravity-driven and dendritic.

- Recent studies in distributed reservoir management have revealed that the placement of reservoirs plays an important role in flood control. Ayalew et al. (2013) develop a framework that combines rainfall-runoff modeling, reservoir routing and Monte Carlo simulation to assess reservoir-regulated flood response [183], and then subsequently use this framework to investigate the effects of reservoir placement on flood frequency [45]. Using a randomly-generated 1,000-year rainfall time series along with a simulated catchment, they find that two retention basins placed in parallel provide better flood control than either (i) two retention basis placed in series along the river main stem, or (ii) a single large retention basin upstream of the watershed outlet. This research demonstrates that placement of hydraulic control structures exerts a powerful influence on the performance of flood control infrastructure, and raises questions about how larger numbers of control structures should best be distributed throughout a watershed to improve flood control.

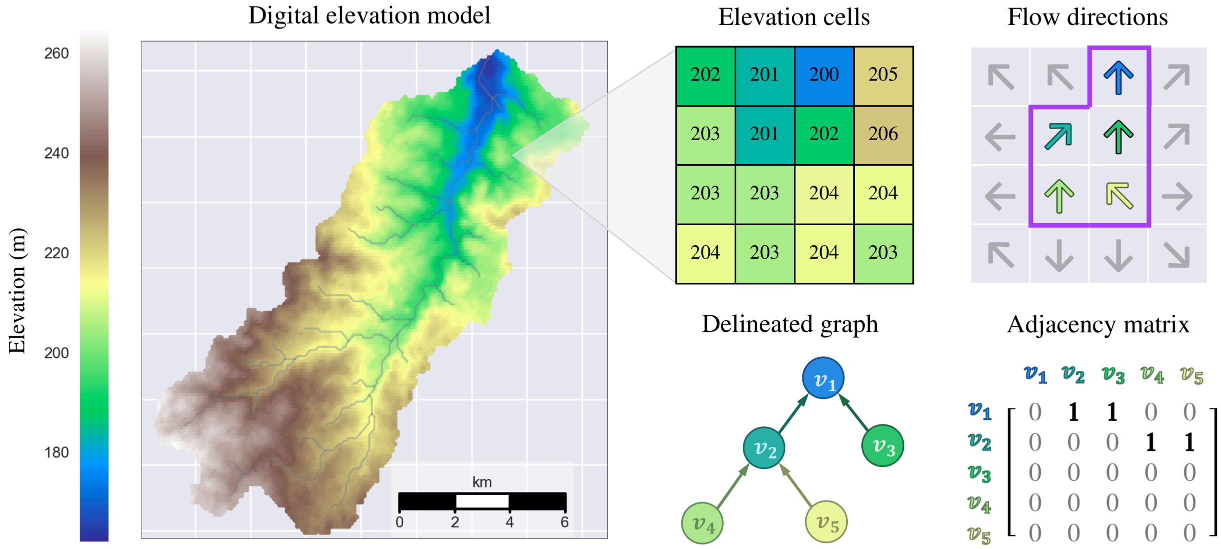
- Inspiration for the controller placement problem can be drawn from recent theoretical work into the controllability of complex networks. These studies show that the control properties of complex systems ranging from power grids to gene expression pathways are inextricably linked with topological properties of an underlying network representation [184]. The location of driver nodes needed for complete controllability of a linear system, for instance, can be determined from the maximum matching of a graph associated with that system's state space representation [185]. For systems in which complete control of the network is infeasible, the relative performance of driver node configurations can be measured by detecting controllable substructures [186], or by leveraging the concept of "control energy" from classical control theory [187–190]. While these studies bring a theoretical foundation to the problem of controller placement, they generally assume linear system dynamics, and may thus not be well-suited for drainage networks, which are driven by nonlinear runoff formation and channel routing processes.

- Recent studies have drawn on advances in complex network theory to examine the controllability of stream networks [49] and enhance understanding of geomorphological processes [191]. Riasi and Yeghiazarian (2017) apply several theoretical controllability metrics to real-world drainage networks, ultimately finding that control of dendritic river networks requires a relatively large proportion of driver nodes [49]. Czuba and Foufoula-Georgiou (2015) investigate spatial and temporal patterns in sediment accumulation in a river network arising from the combined effects of transport dynamics and stream network topology [191]. They find that the emergence of persistent clusters of mass on the network is a major driver of geomorphological change, and conclude that management efforts should seek to "identify the source contributions that synchronize on the network to form clusters", and then break the synchronization by reducing sediment generation in these regions.

Despite the critical need for system-scale stormwater control, there is to our knowledge no robust theoretical framework to guide the placement of hydraulic control structures for the purposes of improving hydrograph peak attentuation. To address this knowledge gap, we formulate a new graph-theoretic algorithm that uses the network structure of watersheds to determine the controller locations that will maximally "de-synchronize" tributary flows. By flattening the discharge hydrograph, our algorithm provides a powerful method to mitigate flash floods and curtail water quality impairments in urban watersheds. Our approach is distinguished by the fact that it is theoretically-motivated, and links the control of stormwater systems with the underlying structure of the drainage network. The result is a fast, generalized algorithm that requires only digital elevation data for the watershed of interest. More broadly, through our graph-theoretic framework we show that network structure plays a dominant role in the control of drainage basins, and demonstrate how the study of watersheds as complex networks can inform more effective stormwater infrastructure design.

6.2 Algorithm description

Flashy flows occur when large volumes of runoff arrive synchronously at a given location in the drainage network. If hydraulic control structures are placed at strategic locations, flood waves can be mitigated by "de-synchronizing" tributary flows before they arrive at a common junction. With this in mind, we introduce a controller placement algorithm that minimizes flashy flows by removing regions of the drainage network that contribute disproportionately to synchronous flows at the outlet. In our approach, the watershed is first transformed into a directed graph consisting of unit subcatchments (vertices) connected by flow paths (edges). Next, critical regions are identified by computing the catchment's *width function* (an approximation of the distribution of travel times to the outlet), and then weighting each vertex in the network in proportion to the number of vertices that share the same travel time to the outlet. The weights are used to compute a *weighted accumulation* score for each vertex, which sums the weights of every possible subcatchment in the watershed. The graph is then partitioned recursively based on this weighted accumulation score, with the most downstream vertex of each partition representing a controller location.



202	201	200	205
203	201	202	206
203	203	204	204
204	203	204	203

$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ v_2 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ v_3 & 0 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 6.1. Watershed as a directed graph. Left panel: Digital elevation model (DEM) of a watershed with river network highlighted. Right panel (from left to right, top to bottom): (i) DEM detail (colors not to scale); (ii) flow directions; (iii) delineated subcatchment graph; (iv) adjacency matrix representation of graph.

Definitions

Graph representation of a watershed: Watersheds can be represented as directed graphs, in which subcatchments (vertices or cells) are connected by elevation-dependent flow paths (edges). The directed graph can be formulated mathematically as an adjacency matrix, A, where for each element a_{ij}, $a_{ij} \neq 0$ if there exists a directed edge connecting vertex v_j to v_i, and conversely, $a_{ij} = 0$ if there does not exist a directed edge connecting vertex v_j to v_i. Nonzero edge weights can be specified to represent travel times, distances, or probabilities of transition between connected vertices. Flow paths between adjacent cells are established using a routing scheme, typically based on directions of steepest descent (see Figure 6.1).

In this study, we determine the connectivity of the drainage network using a *D8 routing* scheme [192]. In this scheme, elevation cells are treated as vertices in a 2-dimensional lattice (meaning that each vertex v_i is surrounded by eight neighbors $\mathcal{N}_i$). A directed link is established from vertex v_i to a neighboring vertex v_j if the slope between v_i and v_j is steeper than the slope between v_i and all of its other neighbors $\mathcal{N}_i \setminus v_j$ (where v_j has a lower elevation than v_i). The *D8 routing* scheme produces a directed acyclic graph where the indegree of each vertex is between 0 and 8 (with an indegree of 8 indicating that the vertex is a "sink"), and the outdegree of each vertex is 1 (except for the watershed outlet, which has an outdegree of 0). It should be noted that other schemes exist for determining drainage network structure, such as the *D-infinity* routing algorithm, which better resolves drainage directions on hillslopes [193]. However, because the routing scheme is not essential to the construction of the algorithm, we focus on the simpler *D8* routing scheme for this study. Similarly, to simplify the construction of the algorithm, we will assume that the vertices of the watershed are defined on a regular grid, such that the area of each unit subcatchment is equal[7]. Figure 6.1 shows the result of delineating a river network from a digital elevation model (left), along with an illustration of the underlying graph structure and adjacency matrix representation (right).

Controller: In the context of this study, a controller represents any structure or practice that can regulate flows from an upstream channel segment to a downstream one. Examples include retention basins, dams, weirs, gates and other hydraulic control structures. These structures may be either passively or actively controlled. For the validation assessment presented later in this paper, we will examine the controller placement problem in the context of *volume capture*, meaning that controllers are passive, and that they are large enough to completely remove flows from their upstream contributing areas. However, the algorithm itself does not require the controller to meet these particular conditions.

Mathematically, we can think of a controller as a cut in the graph that removes one of the edges. This cut halts or inhibits flows across the affected edge. Because the

[7] Thus, for watershed models derived from a digital elevation model (DEM), a unit subcatchment is equivalent to a single DEM cell.

watershed has a dendritic structure, any cut in the network will split the network into two sub-trees: (i) the delineated region upstream of the cut, and (ii) all the vertices that are not part of the delineated region. Placing controllers is thus equivalent to removing branches (subcatchments) from a tree (the parent watershed).

Delineation: Delineation returns the set of vertices upstream of a target vertex. In other words, this operation returns the contributing area of vertex v_i. Expressed in terms of the adjacency matrix:

$$V_d(A, v_i) = \{v_j \in V | (A^n)_{ij} \neq 0 \text{ for some } n \leqslant D\} \tag{6.1}$$

Where A^n is the adjacency matrix A raised to the n^{th} power, i is the row index, j is the column index, V is the vertex set of A, and D is the graph diameter. Note that $(A^n)_{ij}$ is nonzero only if vertex v_j is located within an n-hop neighborhood of vertex v_i. Note that the delineation operation can also be performed in a single step by analyzing the null space of the graph Laplacian of the watershed's adjacency matrix [171].

Pruning: Pruning is the complement of delineation. This operation returns the vertex set consisting of all vertices that are not upstream of the current vertex.

$$V_p(A, v_i) = V \setminus V_d(A, v_i) \tag{6.2}$$

Subgraphs induced by the delineated and pruned vertex sets are defined as follows:

$$\begin{aligned} A_d(A, v_i) &= A(G[V_d]) \\ A_p(A, v_i) &= A(G[V_p]) \end{aligned} \tag{6.3}$$

Where $A(G[V])$ represents the adjacency matrix of the subgraph induced by the vertex set V.

Width function: The width function describes the distribution of travel times from each upstream vertex to some downstream vertex, v_i[8] [195]. In general terms, the width

[8] The width function $H(x)$ was originally defined by Shreve (1969) to yield the number of links in the network at a topological distance x from the outlet [194]. Because travel times may vary between hillslope and channel links, we present a generalized formulation of the width function here.

function can be expressed as:

$$H(t, v_i) = \sum_{\gamma \in \Gamma_i} I(\gamma, t) \tag{6.4}$$

Along with an indicator function, $I(\gamma, t)$:

$$I(\gamma, t) = \begin{cases} 1 & T(\gamma) = t \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

Where Γ_i is the set of all directed paths to the target vertex v_i, and $T(\gamma)$ is the travel time along path γ. If the travel times between vertices are constant and equal for all vertices, the width function of the graph at vertex v_i can be described as a linear function of the adjacency matrix:[9]

$$H(t, v_i) = (A^t \mathbf{1})(i) \tag{6.6}$$

Where $\mathbf{1}$ signifies the vector of all ones, A^t represents the adjacency matrix A raised to the power t (with t representing the discrete time step), and $(A^t\mathbf{1})(i)$ indicates the i^{th} element of the vector $A^t\mathbf{1}$. In real-world drainage networks, travel times between grid cells are not uniform. Crucially, the travel time for channelized cells will be roughly 1-2 orders of magnitude faster than the travel time in hillslope cells [195, 196]. Thus, to account for this discrepancy, we define ϕ to represent the ratio of hillslope to channel travel times:

$$\phi = \frac{t_h}{t_c} \tag{6.7}$$

Where t_h is the travel time for hillslopes and t_c is the travel time for channels. Figure 6.2 (left) shows the width function for an example watershed, under the assumption that channel velocity is ten times faster than hillslope velocity ($\phi = 10$). The width functions for various values of ϕ are shown in Figures S6.3 and S6.4 in the Supplementary Information.

Note that when the effects of hydraulic dispersion are ignored, the width function is equivalent to the geomorphological impulse unit hydrograph (GIUH) of the basin [195].

[9] While mathematically concise, this equation is computationally inefficient. See Section S6.1 in the Supplementary Information for the efficient implementation used in our analysis.

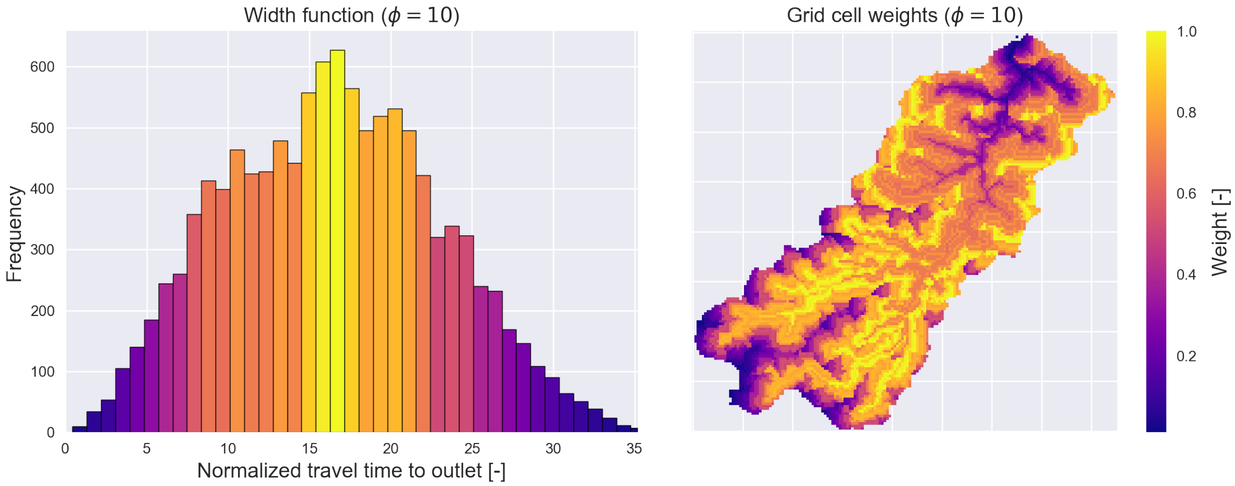


Figure 6.2. Width function and vertex weights. Left: width function (travel-time histogram) of the watershed, assuming that channelized travel time is ten times faster than on hillslopes ($\phi = 10$). Right: weights associated with each vertex of the graph. Brighter regions correspond to areas that contribute to the peaks of the width function.

The GIUH represents the response of the basin to an instantaneous impulse of rainfall distributed uniformly over the catchment; or equivalently, the probability that a particle injected randomly within the watershed at time $t = 0$ exits the watershed through the outlet at time $t = t'$.

Accumulation: The accumulation at vertex v_i describes the number of vertices located upstream of v_i (or alternatively, the upstream area [197]). It is equivalent to the cumulative sum of the width function with respect to time[10]:

$$C(v_i) = (\sum_{t=0}^{\infty} A^t \mathbf{1})(i) \tag{6.8}$$

Figure 6.3 (left) shows the accumulation at each vertex for an example catchment. Because upstream area is correlated with mean discharge [195], accumulation is frequently used to determine locations of channels within a drainage network [197].

Weighting function: To identify the vertices that contribute most to synchronous flows at the outlet, we propose a weighting function that weights each vertex by its rank in the travel time distribution. Let τ_{ij} represent the known travel time from a starting vertex v_j to the outlet vertex v_i. Then the weight associated with vertex v_j can be expressed

[10] See Section S6.1 in the Supplementary Information for the efficient implementation of the accumulation algorithm.

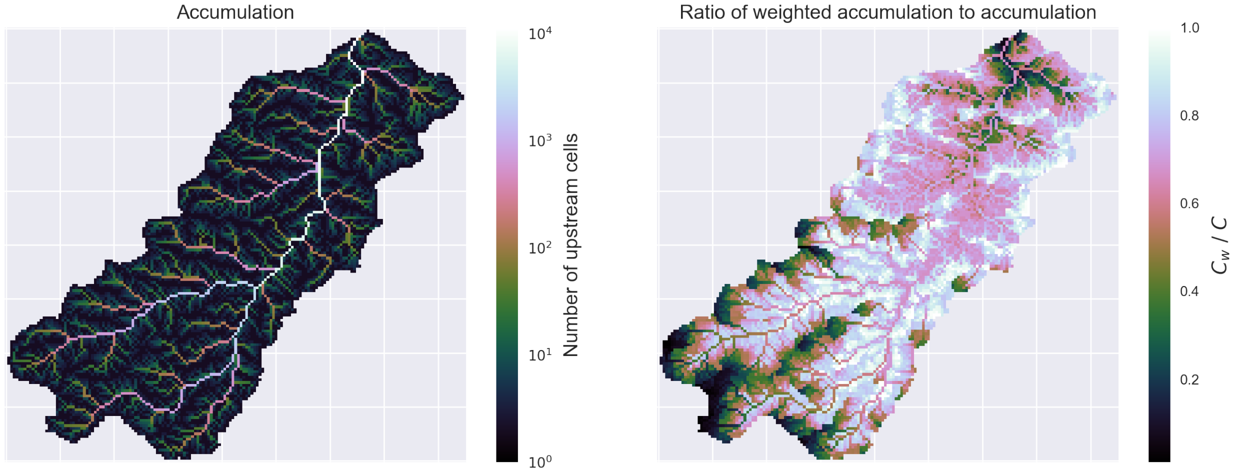


Figure 6.3. Accumulation and weighted accumulation. Left: accumulation (number of cells upstream of every cell). Right: ratio of weighted accumulation to accumulation (C_w/C).

in terms of a weighting function $W(v_i, v_j)$:

$$w_j = W(v_i, v_j) = \frac{H(\tau_{ij}, v_i)}{\max\limits_t(H(v_i))} \tag{6.9}$$

Where τ_{ij} represents the travel time from vertex v_j to vertex v_i, $H(\tau_{ij}, v_i)$ represents the width function for an outlet vertex v_i evaluated at time τ_{ij}, and the normalizing factor $\max\limits_t(H(v_i))$ represents the maximum value of the width function over all time steps t. In this formulation, vertices are weighted by the rank of the associated travel time in the width function. Vertices that contribute to the maximum value of the width function (the mode of the travel time distribution) will receive the highest possible weight (unity), while vertices that contribute to the smallest values of the width function will receive small weights. In other words, vertices will be weighted in proportion to the number of vertices that share the same travel time to the outlet. Figure 6.2 shows the weights corresponding to each bin of the travel time distribution (left), along with the weights applied to each vertex (right). Weights for varying values of ϕ are shown in Figures S6.3 and S6.4 in the Supplementary Information.

Weighted accumulation: Much like the *accumulation* describes the number of vertices upstream of each vertex v_i, the *weighted accumulation* yields the sum of the weights upstream of v_i. If each vertex v_j is given a weight w_j, the weighted accumulation at vertex v_i can be defined:

$$C_w(v_i, \mathbf{w}) = (\sum_{t=0}^{\infty} A^t \mathbf{w})(i) \tag{6.10}$$

Where $\mathbf{w}$ is a vector of weights, with each weight w_j associated with a vertex v_j in the graph. When the previously-defined weighting function is used, the weighted accumulation score measures the extent to which a subcatchment delineated at vertex v_i contributes to synchronous flows at the outlet. In other words, if the ratio of weighted accumulation to accumulation is large for a particular vertex, this means that the subcatchment upstream of that vertex contributes disproportionately to the peak of the hydrograph. Figure 6.3 (right) shows the ratio of weighted accumulation to accumulation for the example catchment. The weighted accumulation provides a natural metric for detecting the cuts in the drainage network that will maximally remove synchronous flows, and thus forms the basis of the controller placement algorithm.

Controller placement algorithm definition

The controller placement algorithm is described as follows. Let A represent the adjacency matrix of a watershed delineated at some vertex v_i. Additionally, let k equal the desired number of controllers, and c equal the maximum upstream accumulation allowed for each controller. The graph is then partitioned according to the following scheme:

1. Compute the width function, $H(t, v_i)$, for the graph described by adjacency matrix A with an outlet at vertex v_i.

2. Compute the accumulation $C(v_j)$ at each vertex v_j.

3. Use $H(t, v_i)$ to compute the weighted accumulation $C_w(v_j)$ at each vertex v_j.

4. Find the vertex v_{opt}, where the accumulation $C(v_{opt})$ is less than the maximum allowable accumulation and the weighted accumulation $C_w(v_{opt})$ is maximized:

$$v_{opt} \leftarrow \underset{v_s \in V_s}{\text{argmax}}(C_w(v_s)) \tag{6.11}$$

   Where V_s is the set of vertices such that vertex v_i is reachable from any vertex in V_s and the accumulation C at any vertex in V_s is less than c.

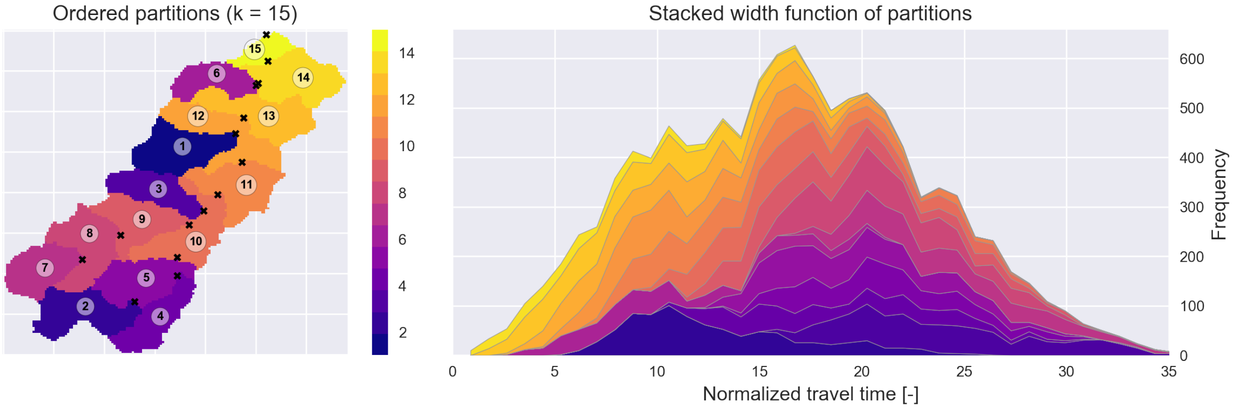


Figure 6.4. Optimal partitions and stacked histogram. Left: partitioning of the example watershed using the controller placement algorithm. Numeric labels indicate the order in which partitions are generated (from first to last). Right: stacked width functions for each partition. The brightness expresses the priority of each partition, with the darker partitions being prioritized over the brighter ones.

5. Prune the graph at vertex v_{opt}: $A \leftarrow A_p(A, v_{\text{opt}})$

6. If the cumulative number of partitions is equal to k, end the algorithm. Otherwise, start at (1) with the catchment described by the new A matrix.

The algorithm is described formally in Algorithm 1. An open-source implementation of the algorithm in the *Python* programming language is also provided [198], along with the data needed to reproduce our results. Efficient implementations of the *delineation*, *accumulation*, and *width function* operations are provided via the `pysheds` toolkit, which is maintained by the authors [199].

Figure 6.4 shows the controller configuration generated by applying the controller placement algorithm to the example watershed, with $k = 15$ controllers, each with a maximum accumulation of $c = 900$ (i.e. each controller captures roughly 8% of the catchment's land area). In the left panel, partitions are shown in order of decreasing priority from dark to light (i.e. darker regions are partitioned first by the algorithm). The right panel shows the stacked width functions for each partition. The sum of the width functions from each partition reconstitute the original width function for the catchment. From the stacked width functions, it can be seen that the algorithm tends to prioritize the pruning of subgraphs that align with the peaks of the travel time distribution. Note for instance, how the least-prioritized paritions gravitate towards the low end of the travel-time distribution, while the most-prioritized partitions are centered around the mode. Controller placement schemes corresponding to different numbers of controllers are shown in Figure S6.5 in the Supplementary Information.

```
Data:
A directed graph described by adjacency matrix $A$;
A target vertex $v_i$ with index $i$;
A desired number of partitions $k$;
The maximum accumulation at each controller, $c$;
Result: Generate partitions for a catchment

Let $\mathbf{q}$ be a vector representing all vertices in the graph;
Let $k_c$ equal the current number of partitions;
Let $ø$ represent a vector of travel times from each vertex to vertex $v_i$;
Let $A$ represent the adjacency matrix of the system;
Let $A_c$ represent the adjacency matrix for the current iteration;

$A_c \leftarrow A$;
$k_c \leftarrow 0$;
while $k_c < k$ do
    $H(t, v_i) \leftarrow (A_c^t \mathbf{1})(i)$;
    $C \leftarrow (\sum_{t=0}^{\infty} A_c^t \mathbf{1})$;
    $\mathbf{w} \leftarrow W(v_i, \mathbf{q})$;
    $C_w \leftarrow (\sum_{t=0}^{\infty} A_c^t \mathbf{w})$;
    if $C(v_i) > 0$ then
        $V_c \leftarrow \{v_m \in V | C(v_m) \leqslant c\}$;
        $V_s \leftarrow V_d(A_c, v_i) \cap V_c$;
        $v_{opt} \leftarrow \underset{v_s \in V_s}{\text{argmax}}(C_w(v_s))$;
        $A_c \leftarrow A_p(A_c, v_{opt})$;
        $k_c \leftarrow k_c + 1$;
    else
    end
end
```

Algorithm 1: Controller placement algorithm

6.3 Algorithm validation

To evaluate the controller placement algorithm, we simulate the controlled network using a hydrodynamic model, and compare the performance to a series of randomized controller placement configurations. Performance is characterized by the "flatness" of the flow profile at the outlet of the watershed, as measured by both the peak discharge and the variance of the hydrograph (i.e. the extent to which the flow deviates from the mean flow over the course of the hydrologic response). To establish a basis for comparison, we simulate a "volume capture" scenario [57], wherein roughly half of the total contributing area is controlled, and each controller completely captures the discharge from its respective upstream area. Additionally, we simulate a "delayed release" scenario in which each controller continuously releases water from a large, bottom-mounted orifice, thereby delaying rather than halting flows from the upstream channel network. These scenarios are chosen as bounding cases, given that most real-world reservoir operation will fall somewhere between these two regimes.

The validation experiment is designed to test the central premises of the controller placement algorithm: that synchronous cells can be identified from the structure of the drainage network, and that maximally capturing these synchronous cells will lead to a flatter overall hydrologic response. If these premises are accurate, we expect to see two results. First, the controller placement algorithm will produce flatter flows than the randomized control trials. Second, the performance of the algorithm will be maximized when using a large number of small partitions. Using many small partitions allows the algorithm to selectively target the highly-weighted cells that contribute disproportionately to the peak of the hydrograph. Conversely, large partitions capture many extraneous low-weight cells that don't contribute to the peak of the hydrograph. In other words, if increasing the number of partitions improves the performance of the algorithm, it not only confirms that the algorithm works for our particular experiment, but also justifies the central premises on which the algorithm is based.

Experimental design

We evaluate controller configurations based on their ability to flatten the outlet hydrograph of a test watershed when approximately 50% of the contributing area is controlled. This test case is chosen because it presents a practical scenario with real-world constraints, and because it allows for direct comparison of many different controller placement strategies. For our test case, we use the Sycamore Creek watershed, a heavily urbanized creekshed located in the Dallas–Fort Worth Metroplex with a contributing area of roughly 83 km^2 (see Figure 6.1). This site is the subject of a long-term monitoring study led by the authors [102], and is chosen for this analysis because (i) it is known to experience issues with flash flooding, and (ii) it is an appropriate size for our analysis—being large enough to capture fine-scale network topology, but not so large that computation time becomes burdensome.

A model of the stream network is generated from a conditioned digital elevation model (DEM) by determining flow directions from the elevation gradient and then assigning channels to cells that fall above an accumulation threshold. Conditioned DEMs and flow direction grids at a resolution of 3 arcseconds (approximately 70 by 90 m) are obtained from the USGS HydroSHEDS database [200]. Grid cells with an accumulation greater than 100 are defined to be channelized cells, while those with less than 100 accumulation are defined as hillslope cells. This threshold is based on visual comparison with the stream network defined in the National Hydrography Dataset (NHD) [201]. Hillslope cells draining into a common channel are aggregated into subcatchments, with a flow length corresponding to the longest path within each hillslope, and a slope corresponding to the average slope over all flow paths in the subcatchment. Percent impervious area and land cover classifications for each subcatchment are obtained from the National Land Cover Database [202], allowing for overland flow velocities to be reasonably approximated (see Section S6.3 in the Supplementary Information). Channel geometries are assigned to each link within the channelized portion of the drainage network. We assume that each stream segment can be represented by a "wide rectangular channel", which is generally accurate for natural river reaches in which the stream width is large compared to the stream depth [54]. To simulate channel width and depth, we assume a power law relation between accumulation and channel size based on an empirical formulation from Moody and Troutman (2002) [203]:

$$\begin{aligned} \omega &= 7.2 \cdot Q^{0.50 \pm 0.02} \\ h &= 0.27 \cdot Q^{0.30 \pm 0.01} \end{aligned} \tag{6.1}$$

Where ω is stream width, h is stream depth, and Q is the mean river discharge. Knowing the width and depth of the most downstream reach, and assuming that the accumulation at a vertex is proportional to the mean flow, we generate channel geometries using the mean parameter values from the above relations. To simulate the effect of floodplain storage and prevent channel overflow, a trapezoidal floodplain section is added to the top of each channel (see Section S6.4 of the Supplementary Information for additional implementation details).

Controllers are implemented as retention basins regulated by outlet structures with controllable orifices. Orifices are mounted on the bottom of each outlet structure and have approximately 10% of the cross-sectional area of the upstream channel section. For the “volume capture” scenario the orifice is left closed, while for the “delayed release” scenario the orifice is left open. Retention basins are sized using a linear relationship between depth and surface area, and are checked against known real-world retention basins to ensure realistic storage capacities (for additional details, see Section S6.4 in the Supplementary Information).

Using the controller placement algorithm, control structures are placed such that approximately 50±3% of the catchment area is captured by storage basins. To investigate the effect of the number of controllers on performance, optimized controller strategies are generated using between $k = 1$ and $k = 35$ controllers. The ratio of hillslope-to-channel travel times is estimated as $\phi = 50$ based on simulations of the catchment under uncontrolled conditions. We compare the performance of our controller placement algorithm to randomized controller placement schemes, in which approximately 50±3% of the catchment area is controlled but the placement of controllers is random. For this comparison assessment, we generate 50 randomized controller placement trials, each using between $k = 1$ and $k = 24$ controllers.[11] For each randomized trial, the maximum and minimum accumulation that can be handled by each controller is selected, then controllers are placed sequentially until 50±3% of the total catchment is upstream of at least one controller. This procedure is similar to the controller placement algorithm, except that at each iteration, the controller is placed at a random candidate cell (i.e. a cell with an accumulation in the appropriate range) instead of the candidate cell with the greatest weighted accumulation (see Section S6.2 in the Supplementary Information for a detailed description of the procedure).

We simulate the hydrologic response using a hydrodynamic model, and evaluate controller placement performance based on the flatness of the resulting hydrograph. To capture the hydrologic response under various rainfall conditions, we simulate small, medium and large rainfall events, corresponding to 11.2, 16.9, and 23.4 mm of rainfall

[11] While the controller randomization code was programmed to use between 1 and 35 controllers, the largest number of controllers achieved was 24. This result stems from the fact that the randomization algorithm struggled to achieve 30+ partitions without selecting cells that fell below the channelization threshold (100 accumulation).

delivered instantaneously over the first five minutes of the simulation. These rainfall volumes are based on the 1-year, 10-year and 100-year design storms (5-minute duration) from intensity-duration-frequency curves for Tarrant county in Texas [204]. A hydrodynamic model is used to simulate the hydrologic response at the outlet by routing runoff through the channel network using the dynamic wave equations [205]. The simulation performance is measured by both the peak discharge and the total variance of the hydrograph. The variance of the hydrograph (which we refer to as "flashiness") is defined as:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(Q_i - \bar{Q})^2 \tag{6.2}$$

Where Q is the discharge, $\bar{Q}$ is the mean discharge in the storm window, and N is the number of data points in the storm window. This variance metric captures the flow's deviation from the mean over the course of the hydrologic response, and thus provides a natural metric for the flatness of the hydrograph. This metric is important for water quality considerations like first flush contamination or streambed erosion—in which the volume of transported material (e.g. contaminants or sediments) depends not only on the maximum discharge, but also on the duration of flow over a critical threshold [87].

Note that the validation experiment is not intended to faithfully reproduce the precise hydrologic response of our chosen study area, but rather, to test the basic premises of the controller placement algorithm. As such, site-specific details—such as soil types and existing infrastructure—have been deliberately simplified. For situations in which these characteristics exert an important influence on the hydrologic response, one may account for these factors by adjusting the inter-vertex travel times used in the controller placement algorithm.

6.4 Results

The controller placement algorithm produces consistently flatter flows than randomized control trials. Figure 6.5 shows the results of the hydraulic simulation assessment in terms of the resulting hydrographs (bottom left), and the overall flashiness and peak discharge of each simulation (right) for the 1-year storm event under the volume capture

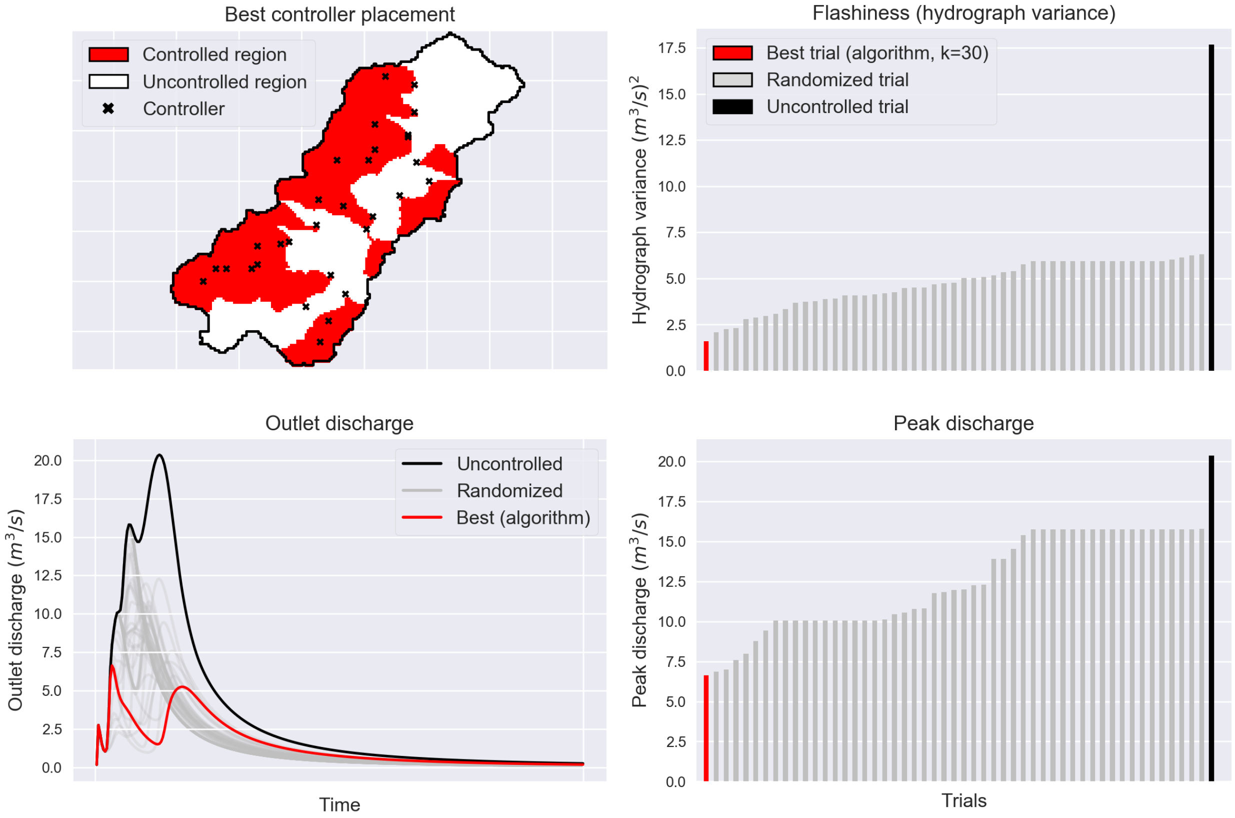


Figure 6.5. Hydraulic modeling results. Results of the hydraulic simulation experiment for the 1-year storm event (11.2 mm) under the volume capture scenario. Top left: best-performing controller placement ($k = 30$), with captured regions in red. Bottom left: hydrographs resulting from each simulation. The uncontrolled simulation is shown in black, while the optimized controller placement simulations are shown in red, and the randomized controller simulations are shown in gray. Right: the overall flashiness (variance of the hydrograph) and peak discharge for each simulation, using the same coloring scheme.

scenario. The best performance is achieved by using the controller placement algorithm with $k = 30$ controllers (see Figure 6.5, top left). Comparing the overall variances and peak discharges, it can be seen that this controller placement produces flatter outlet discharges than any of the randomized controller placement strategies.[12] Specifically, the best controller placement predicted by the algorithm achieves a peak discharge that is roughly 29% of that of the uncontrolled case, while the randomized simulations by comparison achieve an average peak discharge that is more than 61% of that of the uncontrolled case. Similarly, the hydrograph variance of the best controller placement predicted by the algorithm is roughly 9.1% of that of the uncontrolled case, compared to

[12] Note that the controller placement algorithm results in a longer falling limb than the randomized trials. This result stems from the fact that the algorithm prioritizes the removal of grid cells that contribute to the peak and rising limb of the hydrograph, while grid cells contributing to the falling limb are ignored. In other words, the controller placement algorithm shifts discharges from the peak of the hydrograph to the falling limb.

27% for the randomized simulations on average. Across all numbers of controllers considered, the controller placement algorithm yields results in approximately 20% lower variance and 15% lower peak discharge on average compared to the randomized placement strategy.

The performance of the controller placement algorithm holds under varying rainfall scenarios and reservoir operating rules. When tested against storm events of different sizes (10-year and 100-year storm events), the controller placement algorithm continues to outperform randomized control trials, with roughly 18% lower hydrograph variance and 13% lower peak discharge over all numbers of controllers considered (see Figures S6.6-S6.9 in the Supplementary Information). Moreover, when tested under the delayed release scenario (in which each reservoir continuously releases water from a bottom-mounted orifice), the controller placement algorithm performs better than under the volume capture scenario. In particular, the algorithm achieves 28% lower hydrograph variance on average, while the best controller placement strategy obtained by the algorithm achieves 80% lower hydrograph variance than the average randomized placement (see Figures S6.10-S6.13 in the Supplementary Information). While the performance of the algorithm holds under different rainfall and reservoir operation scenarios, it should be noted that the within-group performance varies with rain event size, which could result from the nonlinearities inherent in wave propagation speed (see the supplementary note in Section S6.7). Thus, while the optimized controller placement still produces flatter flows than randomized control trials, the performance of the controller placement algorithm could be further improved by tuning the assumed inter-vertex travel times to correspond to the expected speed of wave propagation.

Under the controller placement algorithm, the best performance is achieved by using a large number of small-scale controllers; however, more controllers does not generally lead to better performance for the randomized controller placement scheme. Given that increasing the number of controllers allows the algorithm to better target highly synchronous cells, this result is consistent with the central premise that capturing synchronous cells will lead to a flatter hydrologic response. Figure 6.6 shows the hydrologic response when the controller placement algorithm is applied using varying numbers of controllers (left), along with associated hydrograph variances (top right) and peak dis-

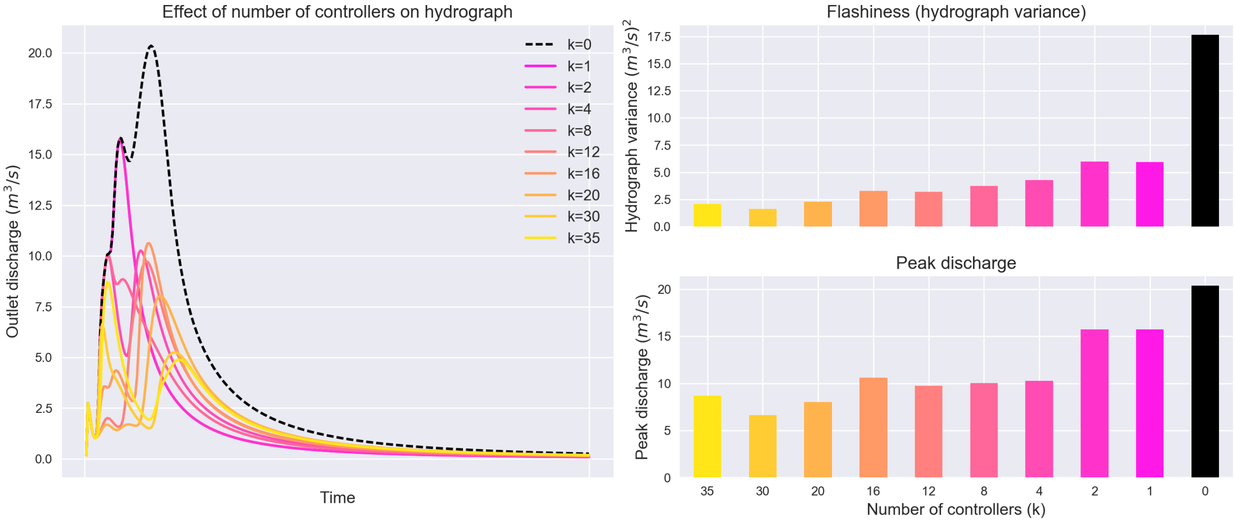


Figure 6.6. Performance vs. number of controllers. Left: hydrographs associated with varying numbers of controllers (k), using the controller placement algorithm with 50% watershed area removal (1-year event, volume capture scenario). Right: hydrograph variance (top) and peak discharge (bottom) vs. number of controllers. In general, more controllers produces a flatter response.

charges (bottom right). In all cases, roughly 50% of the watershed is controlled; however, configurations using many small controllers consistently perform better than configurations using a few large controllers. This trend does not hold for the randomized controller placement strategy (see Figures S6.14 and S6.14 in the Supplementary Information). Indeed, the three best-performing randomized controller placements use a median of $k = 17$ controllers, while the three worst-performing randomized controller placements use a median of $k = 10$ controllers (where performance is measured in terms of the hydrograph variance). By comparison, when the controller placement algorithm is used, the three best-performing simulations use a median of $k = 30$ controllers, while the three worst-performing simulations use a median of $k = 2$ controllers. The finding that the controller placement algorithm converges to a (locally) optimal solution follows from the fact that as the number of partitions increases, controllers are better able to capture highly-weighted regions without also capturing extraneous low-weight cells. This in turn implies that the weighting scheme used by the algorithm accurately identifies the regions of the watershed that contribute disproportionately to synchronized flows. Thus, in spite of various sources of model and parameter uncertainty, the experimental results confirm the central principles under which the controller placement algorithm operates: namely, that synchronous regions can be deduced from the graph structure alone, and

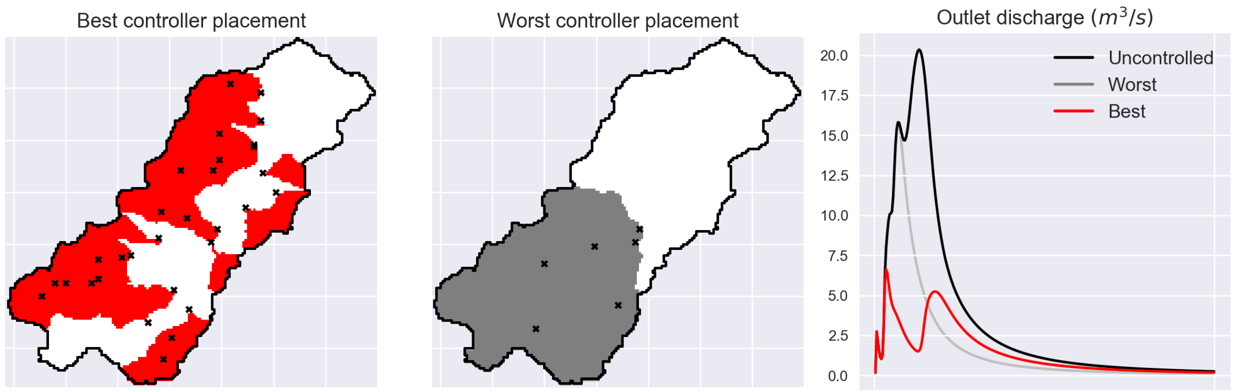


Figure 6.7. Intuition behind controller placement algorithm. Left: Best controller placement in terms of peak discharge ($k = 30$ controllers, using the controller placement algorithm). Center: worst controller placement in terms of peak discharge ($k = 6$ controllers, randomized). Controller locations are indicated by black crosses, and controlled partitions are indicated by colored regions. Right: hydrographs associated with the best and worst controller placement strategies (1-year storm event, volume capture scenario).

that controlling these regions results in a flatter hydrograph compared to randomized control trials.

In addition to demonstrating the efficacy of the controller placement algorithm, the validation experiments reveal some general principles for organizing hydraulic control structures within drainage networks to achieve downstream streamflow objectives. Overall, the controller placement strategies that perform best—whether achieved through optimization or randomization—tend to partition the watershed axially rather than laterally. These lengthwise partitions result in a long, thin drainage network that prevents tributary flows from "piling up". Figure 6.7 shows the partitions corresponding to the best-performing and worst-performing controller placement strategies with respect to peak discharge (left and center, respectively), along with the associated hydrographs (right). While the best-performing controller placement strategy evenly distributes the partitions along the length of the watershed, the worst-performing controller placement strategy controls only the most upstream half of the watershed. As a result, the worst-performing strategy removes the largest part of the peak, but completely misses the portion of the peak originating from the downstream half of the watershed. In order to achieve a flat downstream hydrograph, controller placement strategies should seek to evenly distribute controllers along the length of the watershed.

6.5 Discussion

The controller placement algorithm presented in this study provides a tool for designing stormwater control systems to better mitigate floods, regulate contaminants, and protect aquatic ecosystems. By reducing peak discharge, optimized placement of stormwater control structures may help to lessen the impact of flash floods. Existing flood control measures often focus on controlling large riverine floods—typically through existing centralized assets, like dams and levees. However, flash floods may occur in small tributaries, canals, and even normally dry areas. For this reason, flash floods are not typically addressed by large-scale flood control measures, despite the fact that they cause more fatalities than riverine floods in the developed world [1]. By facilitating distributed control of urban flash floods, our controller placement strategy could help reduce flash flood related mortality. Moreover, by flattening the hydrologic response, our controller placement algorithm promises to deliver a number of environmental and water quality benefits, such as decreased first flush contamination [87], decreased sediment transport [206], improved potential for treatment in downstream green infrastructure [19, 102], and regulation of flows in sensitive aquatic ecosystems [207].

Key features of the algorithm

The controller placement algorithm satisfies a number of important operational considerations:

- **Theoretically motivated**. The controller placement algorithm has its foundation in the theory of the geomorphological impulse unit hydrograph—a relatively mature theory supported by a established body of research [164–169, 195]. Moreover, the algorithm works in an intuitive way—by recursively removing the subcatchments of a watershed that contribute most to synchronized flows. This theoretical basis distinguishes our algorithm from other strategies that involve exhaustive optimization or direct application of existing graph theoretical constructs (such as graph centrality metrics).

- **Generalizable and extensible**. Because it relies solely on network topology, the controller placement algorithm will provide consistent results for any drainage network, including both natural stream networks and constructed sewer systems. Moreover, because each step in our algorithm has a clear meaning in terms of the underlying hydrology, the algorithm can be modified to satisfy more complex control problems (such as systems in which specific regulatory requirements must be met).

- **Flexible to user objectives and constraints**. The controller placement algorithm permits specification of important practical constraints, such as the drainage area that each control site can capture, and the number of control sites available. Moreover, the weighting function can be adjusted to optimize for a variety of objectives (such as the overall "flatness" of the hydrograph, or removal of flows from a contaminated upstream region).
- **Parsimonious with respect to data requirements**. The controller placement algorithm requires only a digital elevation model of the watershed of interest. Additional data—such as land cover and existing hydraulic infrastructure—can be used to fine-tune estimates of travel times within the drainage network, but are not required by the algorithm itself.
- **Fast implementation** For the watershed examined in this study (consisting of about 12,000 vertices), the controller placement algorithm computes optimal locations for $k = 15$ controllers in roughly 3.0 seconds (on a 2.9 GHz Intel Core i5 processor). While the computational complexity of the algorithm is difficult to characterize[13], it is faster than other comparable graph-cutting algorithms, such as recursive spectral bisection or spectral clustering, both of which are $O(n^3)$ in computational complexity.

Taken together, our algorithm offers a solution to the controller placement problem that is suitable for research as well as for practical applications. On one hand, the algorithm is based in hydrologic and geomorphological theory, and provides important insights into the connections between geomorphology and the design of the built environment. On the other hand, the algorithm is fast, robust, and easy-to-use, making it a useful tool for practicing engineers and water resource managers.

[13] The computational complexity of the controller placement algorithm depends on the implementation of component functions (such as delineation and accumulation computation), which can in turn depend on the structure of the watershed itself.

Caveats and directions for future research

While our controller placement algorithm is robust and broadly-applicable, there are a number of important considerations to keep in mind when applying this algorithm to real-world problems.

- The controller placement algorithm implicitly assumes that rainfall is uniform over the catchment of interest. While this assumption is justified for small catchments in which the average spatial distribution of rainfall will be roughly uniform, this assumption may not hold for large (e.g. continent-scale) watersheds. Modifications to the algorithm would be necessary to account for a non-uniform spatial distribution of rainfall.
- The controller placement algorithm is sensitive to the chosen ratio of hillslope to channel speeds, ϕ. Care should be taken to select an appropriate value of ϕ based on site-specific land cover and morphological characteristics. More generally, for situations in which differential land cover, soil types, and existing hydraulic infrastructure play a dominating role, the performance of the algorithm may be enhanced by adjusting inter-vertex travel times to correspond to estimated overland flow and channel velocities.
- Our assessment of the algorithm's performance rests on the assumption that installed control structures (e.g. retention basins) are large enough to capture upstream discharges. The algorithm itself does not explicitly account for endogenous upstream flooding that could be introduced by installing new control sites.
- In this study, experiments were conducted only for impulsive rainfall inputs (i.e. with a short duration of rainfall). Future work should assess the performance of the distance-weighted controller placement strategy under arbitrary rainfall durations.
- Our analysis assumes that reservoirs are initially empty before each storm event. While some previous studies in distributed reservoir operation contend that an initially-empty condition is "the simplest and most defensible approach" [208], other studies use random initialization of reservoir depths to simulate the effect of successive storm events when reservoir operation rules are unknown [183]. While this

latter approach may provide more realistic results under unknown reservoir operating conditions, we ultimately use an initially-empty condition due to the combinatorial difficulty of assessing the effect of random initial depths alongside varying numbers of controllers, controller placement strategies, and rainfall scenarios. Random initialization could potentially effect the results by inducing overflows under the volume capture scenario, or by quickening the rising limb under the delayed release scenario. Additional work is needed to understand how these effects would impact the performance of the controller placement algorithm compared to randomized control trials. With this in mind, it should be noted that new "smart" water systems are enabling more flexible control of distributed stormwater infrastructure [19], which may in turn strengthen the assumption of initially-empty storage conditions. Bartos et al. (2018), for instance, present a real-world case study in which real-time analytics and control are used to pre-emptively evacuate retention basins before a storm event, reducing the magnitude of the downstream hydrologic response [102]. Because initially emptying storage basins often leads to a favorable hydrologic response [106], assuming empty or near-empty initial storage conditions may be more realistic than assuming random initial depths for systems with real-time control capabilities.

- This study focuses primarily on event-based diagnostics of system performance—specifically, by measuring the flatness of the hydrologic response under independent storm events. However, it should be noted that water infrastructure may also be evaluated in terms of long-term performance—for instance, by measuring the response of the system to an extended stochastic rainfall time series. While not computationally feasible for the model used in this study, future work should investigate the performance of the controller placement algorithm under extended hydrodynamic simulations.

More broadly, future research should investigate the problem of sensor placement in stream networks using the theoretical framework developed in this paper. While this study focuses on the problem of placement of hydraulic control structures, our algorithm also suggests a solution to the problem of sensor placement. Stated probabilistically, the geomorphological impulse unit hydrograph (GIUH) represents the probability that a "particle" injected randomly within the watershed at time $t = 0$ exits the outlet at time $t = t'$. Thus, the peaks of the GIUH correspond to the portions of the hydrologic response where there is the greatest amount of ambiguity about where a given "particle" originated. It follows that the same locations that maximally de-synchronize flows may also be the best locations for disambiguating the locations from which synchronous flows originated. Future experiments should investigate the ability to estimate upstream states (e.g. flows) within the network given an outlet discharge along with internal state observers (e.g. flow sensors) placed using the algorithm developed in this study.

6.6 Conclusions

We develop an algorithm for placement of hydraulic control structures that maximally flattens the hydrologic response of drainage networks. This algorithm uses the geomorphological impulse unit hydrograph to locate subcatchments that dominate the peaks of the hydrograph, then partitions the drainage network to minimize the contribution of these subcatchments. We find that the controller placement algorithm produces flatter hydrographs than randomized controller placement trials—both in terms of peak discharge and overall variance. By reducing the flashiness of the hydrologic response, our controller placement algorithm may one day help to mitigate flash floods and restore urban water quality through reduction of contaminant loads and prevention of streambed erosion. We find that the performance of the algorithm is enhanced when using a large number of small, distributed controllers. In addition to confirming the central hypothesis that synchronous cells can be identified based on network structure of drainage basins, this result lends justification to the development of decentralized *smart* stormwater systems, in which active control of small-scale retention basins, canals and culverts enables more effective management of urban stormwater. Overall, our algorithm is efficient, requires only digital elevation model data, and is robust to parameter and model uncertainty, making it suitable both as a research tool, and as a design tool for practicing water resources engineers.

Published version

This chapter was reprinted with permission from the article [209]:
Bartos M, Kerkez B. (2019) *Adv. Water Resour.*, **127**, 167-179

Acknowledgments

Funding for this project was provided by the National Science Foundation (Grants 1639640 and 1442735) and the University of Michigan. We would like to thank Alex Ritchie for exploring alternative approaches to the controller placement problem and for his help with literature review. We would also like to thank Dr. Alfred Hero for his advice in formulating the problem.

Declarations of interest

Declarations of interest: none.

Data availability

Code and data links are available at:
`https://github.com/kLabUM/hydraulic-controller-placement`

Supplementary Information

S6.1 Implementations of algorithms used in the study

Width function

For the software implementation used in this study, the width function is computed by determining the travel times from each vertex to the outlet, and then generating a binned histogram of these travel times. The travel times from each vertex to the outlet are computed by performing a depth-first search on the graph representation of the watershed starting with the outlet, and then recording the distances from each vertex to the outlet. The travel times are then binned to produce the width function. The travel time computation is implemented as **grid.flow_distance** in the *pysheds* software package, available at `github.com/mdbartos/pysheds`.

Note that if matrix multiplication is used to compute the width function, the inter-vertex travel times cannot be used as the weights of the adjacency matrix. Rather, differential travel times may be accounted for by modifying the topology of the graph. For instance, consider a graph consisting of fast nodes and slow nodes, where fast nodes transfer flow 10 times as quickly as slow nodes. In this scheme, slow nodes can be modeled using 10 “dummy” vertices placed in series. It should be noted however, that this implementation is inefficient both in terms of speed of computation and memory usage.

Flow accumulation

The vectorized flow accumulation algorithm (developed previously by the authors, but unpublished) is reproduced here for the reader's convenience:

1. Create an m x n array, **edges** to represent the edges of the directed graph. For each entry in the array, the index corresponds to the index of the start node, and the value corresponds to the index of the end node.
2. Create an m x n array, **in_degree**, to hold the in-degree of each grid cell (i.e. the number of cells currently pointing to that cell). This can be accomplished by counting the number of occurrences of each unique value in **edges**.
3. Create an m x n array of ones **flow_accumulation** to hold the computed number of upstream cells for each cell.
4. Define a 1 x nm array **startnodes** with entries equal to the indices of **edges**.
5. Define a 1 x nm array **endnodes** with entries equal to the values of **edges**.
6. Create a 1 x nm boolean index **no_pred** which is 0 where **in_degree** is greater than 0, and 1 where **in_degree** is equal to zero.
7. Select the subset of start nodes and end nodes that have no predecessors: **startnodes = startnodes[no_pred]** and **endnodes = endnodes[no_pred]**. This selects the "outermost layer" of nodes.
8. While **endnodes** is not empty:
   - Add the flow accumulation at the start nodes to the flow accumulation of the end nodes:
     - **flow_accumulation[endnodes] += flow_accumulation[startnodes]**
   - Decrement the in-degree of the endnodes by the number of start nodes that are linked to it in the current step. With **endnodes** containing repeated entries this operation can be represented as: **in_degree[endnodes] -= 1**.
   - Set the new value of **startnodes** as the unique elements in **endnodes** with a current in-degree of zero: **startnodes = unique(endnodes)[(in_degree == 0)]**

- Set **endnodes** as the end nodes corresponding to the new start nodes: **endnodes = edges[startnodes]**.

This algorithm is implemented as **grid.accumulation** in the *pysheds* software package, available at `github.com/mdbartos/pysheds`.

S6.2 Randomized controller placement procedure

In this section, we describe the randomized controller placement procedure. Before running the randomized controller placement procedure, the following parameters must be specified:

- A target number of controllers, k.
- A tolerance parameter, γ, which determines the range of contributing areas that each controller can handle.
- The total number of cells, $n \pm \epsilon$ to be captured by all controllers. In our experiment, this parameter is set to equal $50 \pm 3\%$ of the total catchment area.

Once these parameters have been selected, the randomized placement procedure proceeds as follows:

1. The accumulation a_i for each vertex in the catchment is computed.
2. Based on the target number of controllers k, and the tolerance parameter, γ, a range of feasible accumulation values $[a_{min}, a_{max}]$ for the controller is selected. For our experiment, this range is defined by $\frac{n}{k} \pm \gamma$.
3. All vertices that have accumulation values within the range $[a_{min}, a_{max}]$ are selected and placed into a list of candidate cells.
4. A random vertex v is selected from this list of candidate cells. The controller is placed at this vertex, and the area upstream of the vertex is removed from the catchment before recomputing the accumulation.
5. If the number of cells captured by all controllers is less than $n - \epsilon$, return to step (1).
6. If the number of cells captured by all controllers is within $[n - \epsilon, n + \epsilon]$, end the procedure and retain the resulting controller placement.
7. If the number of cells captured by all controllers is greater than $n \pm \epsilon$, abort the procedure and start over.

S6.3 Determination of land cover information

Percent impervious area and land cover classifications are retrieved from the National Land Cover Database (NLCD). Next, estimated Manning's n values for each NLCD class are mapped to corresponding pixels in the land cover dataset [210]. The NLCD datasets are then clipped to the catchment of interest using the `pysheds` library and then used in the generation of subcatchments in the hydrodynamic model. Visualizations of impervious area and estimated Manning's n values are shown in Figure S6.1.

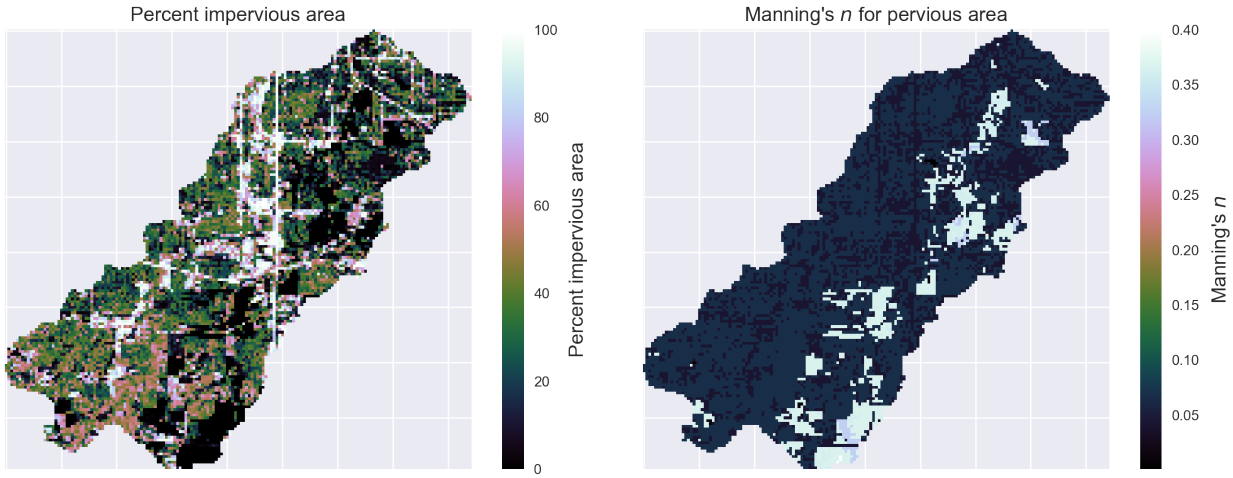


Figure S6.1. Impervious area and Manning's n. Determination of percent impervious area (left) and estimated Manning's n for pervious subareas.

S6.4 Description of channel geometry and storage elements

Channel geometry is modeled using a wide rectangular channel section along with a trapezoidal floodplain section attached to the top of the channel to capture overflow. A schematic of the channel geometry is shown in Figure S6.2 (left). The width (w) and depth (d) of the rectangular section are computed based on contributing area using the hydraulic geometry relations specified in the main paper. The width and depth of the trapezoidal section are computed by scaling the width and depth of the main channel. Given no prior information, we take $\alpha = 1$ and $\beta = 1.5$. Because the floodplain section rarely fills except for the case of channel sections located upstream of storage elements during large storm events, these parameter values have a minimal effect on the downstream hydrograph.

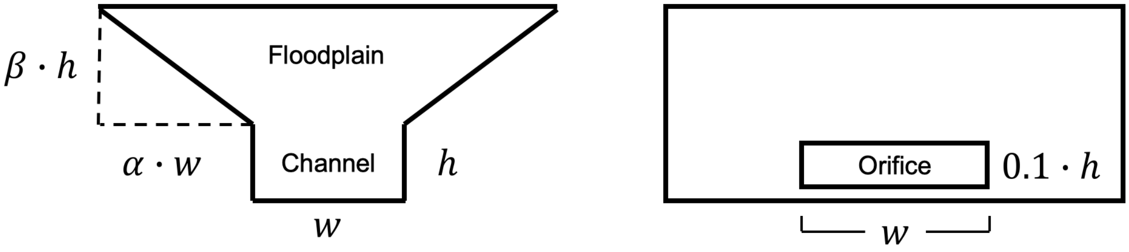


Figure S6.2. Channel and storage geometry. Channel geometry (left) and geometry of outlet structure (right).

Controllers are implemented as storage ponds throttled by controllable orifices. Orifices are positioned at the bottom of each outlet structure, and their cross-sectional areas are set at approximately 10% of the cross sectional area of the upstream channel section, as shown in Figure S6.2 (right). This large cross-sectional area is chosen to differentiate the "delayed release" scenario from the "volume capture scenario". For the "volume capture" scenario, the orifice is completely closed, while for the "delayed release" scenario, the orifice is completely open. The elevation of the bottom of the storage pond is taken to be equal to the elevation of the upstream channel bottom. The storage curve is specified in terms of a functional relationship between depth and surface area:

$$S = 7.5e4 \left(\frac{\mathrm{m}^2}{\mathrm{m}}\right) \cdot \mathrm{h} + 1000\ \mathrm{m}^2 \tag{6.1}$$

Where S represents the surface area of the pond in meters, and h represents the depth of the pond in meters. This relation was chosen based on its ability to prevent flooding during the 1-controller, 100-year storm scenario. It should be noted that this functional relationship has no effect on the results of the "volume capture" scenario, in which all runoff upstream of the storm event is captured.

Storage volumes are found to be realistic for all scenarios considered. The largest storage volumes are required under the single-controller scenario. Under this scenario, the storage volumes required for the 1-year, 10-year and 100-year storm events are approximately 160,000 m^3, 270,000 m^3 and 400,000 m^3, respectively. For the more realistic 30-pond scenario, the largest storage volumes required are 10,000 m^3, 16,000 m^3, and 22,000 m^3 for the same set of rainfall scenarios. While there is little public information available regarding typical retention basin sizes, we can compare these volumes to known retention basin volumes from the city of Ann Arbor. The largest stormwater retention basin in the city of Ann Arbor has a capacity of roughly 70,000 m^3, and drains an area roughly one-third the size of the catchment used in this study (27 km^2). Thus, the storage volumes used in our experiments appear to be reasonable.

S6.5 Vertex weights as a function of ϕ

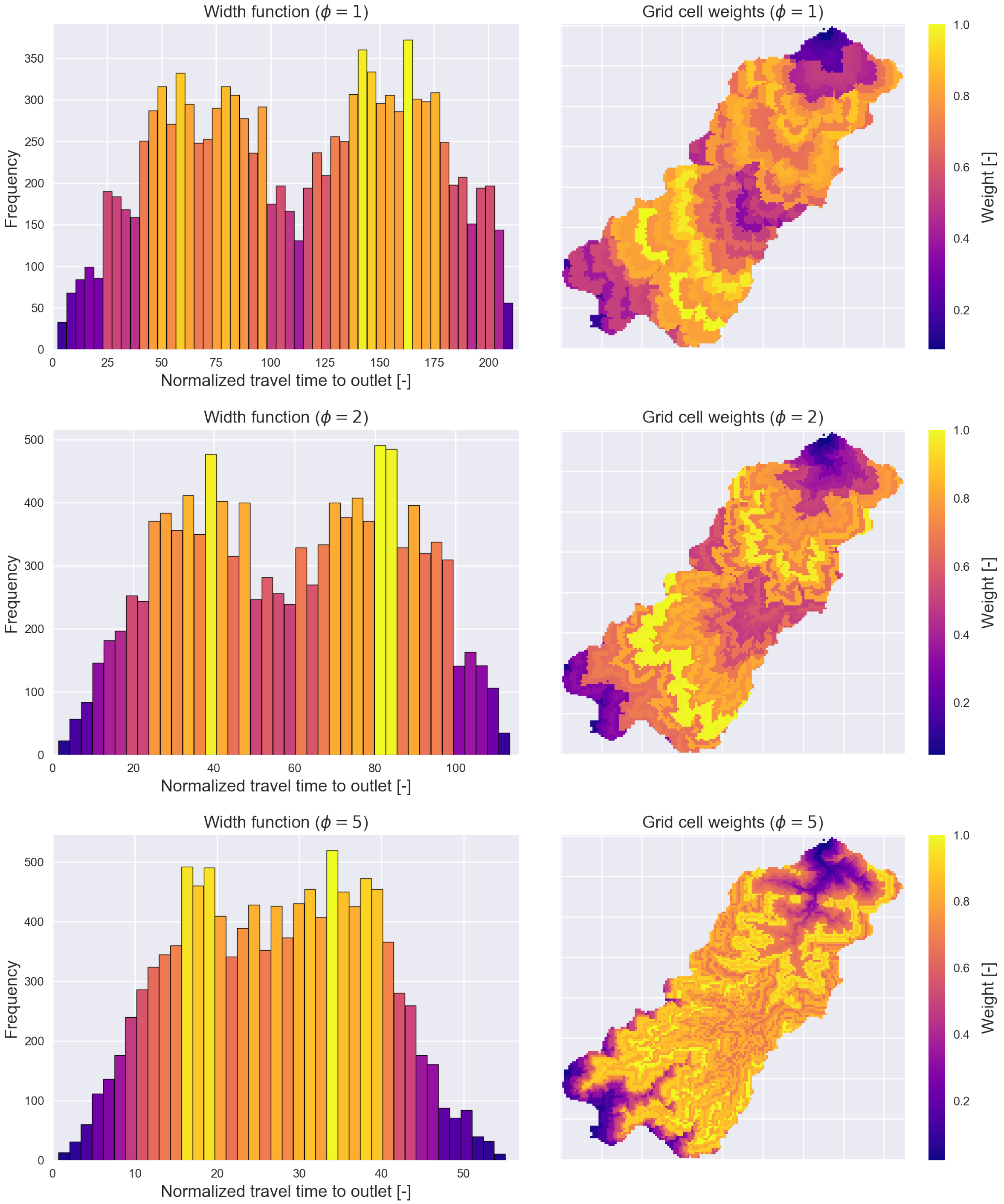


Figure S6.3. Left: width function. Right: vertex weights.

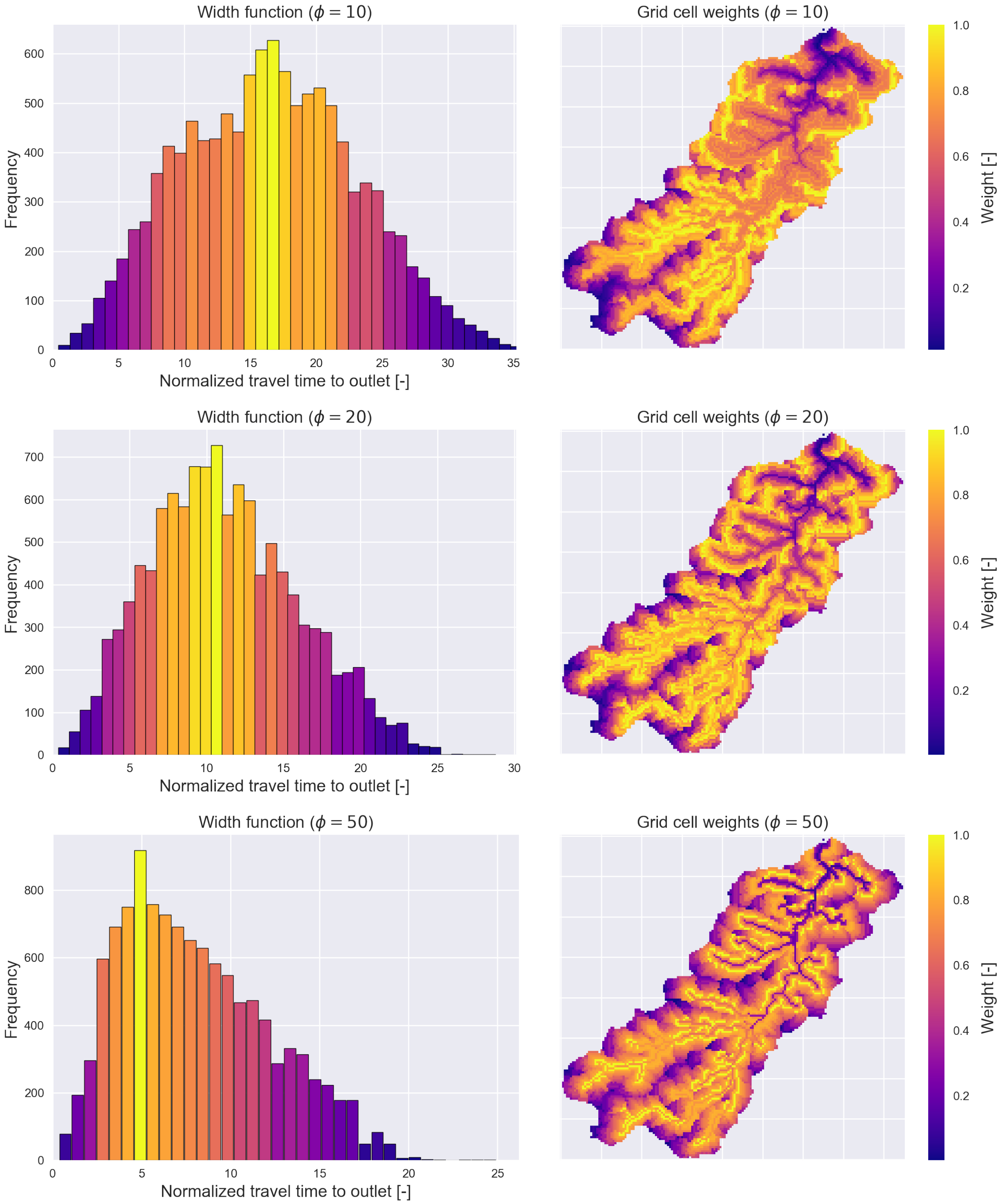


Figure S6.4. Left: width function. Right: vertex weights.

S6.6 Optimal controller placements for various numbers of controllers (k)

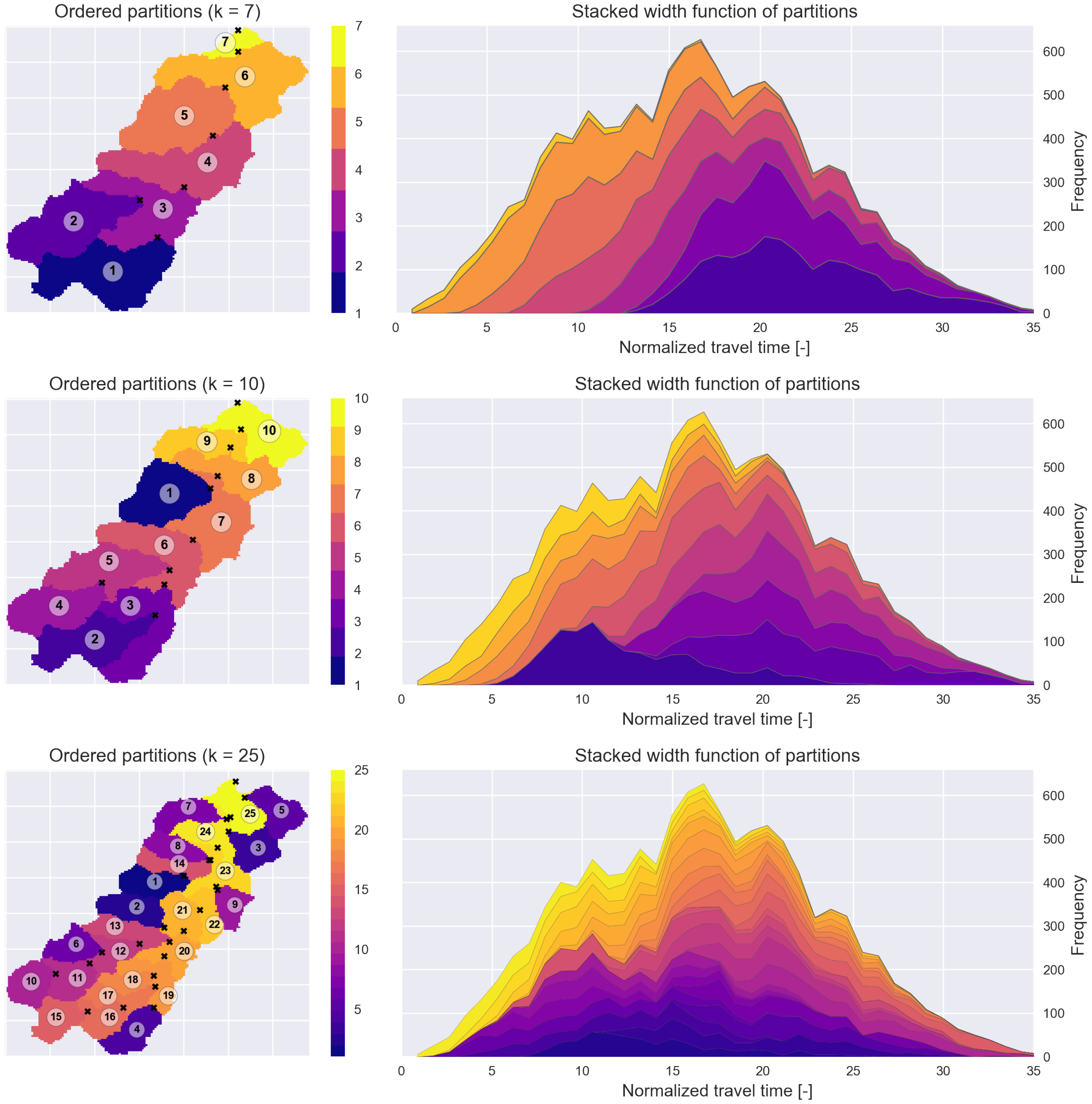


Figure S6.5. Partitions and stacked width functions for varying numbers of controllers. Optimized controller placements (left) and stacked width functions (right) for varying number of controllers $k \in \{7, 10, 25\}$, $\phi = 10$.

S6.7 Performance under rainfall events of different sizes

When tested against storm events of different sizes, the controller placement algorithm generally outperforms randomized control trials. However, the relative performance between simulations varies with rainfall intensity, which suggests that a uniquely optimal controller placement may not exist for rainfall events of all sizes. As shown in Figures S6.6 and S6.7, the optimized controller placement still produces flatter flows overall compared to randomized trials for the small and large storm events. However, for the large storm event, one of the randomized simulations produces a slightly smaller peak discharge than the best-performing optimized controller placement. Moreover, the within-group performance of controller placement strategies varies with storm event size, as seen in Figures S6.8 and S6.9. For instance, the controller placement that produces the smallest peak discharge under the large storm event produces the 3rd smallest peak discharge under the medium storm event, and the 5th smallest peak discharge under the small storm event (Figure S6.9). These results suggest that the optimal controller placement for large storms may not be the same as the optimal controller placement for small storms. This situation may result from the fact that larger flood waves travel faster, meaning that inter-vertex travel times will change depending on the scale of the hydrologic response. Consequently, assumed inter-vertex travel times (controlled in this experiment by the parameter ϕ) may need to be tuned depending on storm event size to account for the nonlinearities inherent in flood wave travel times. Despite these parameter selection issues, the experiments show that controller placement algorithm still produces flatter flows overall than random control trials for storm events of various sizes.

S6.8 Hydrograph variance for storm events of different sizes under volume capture scenario

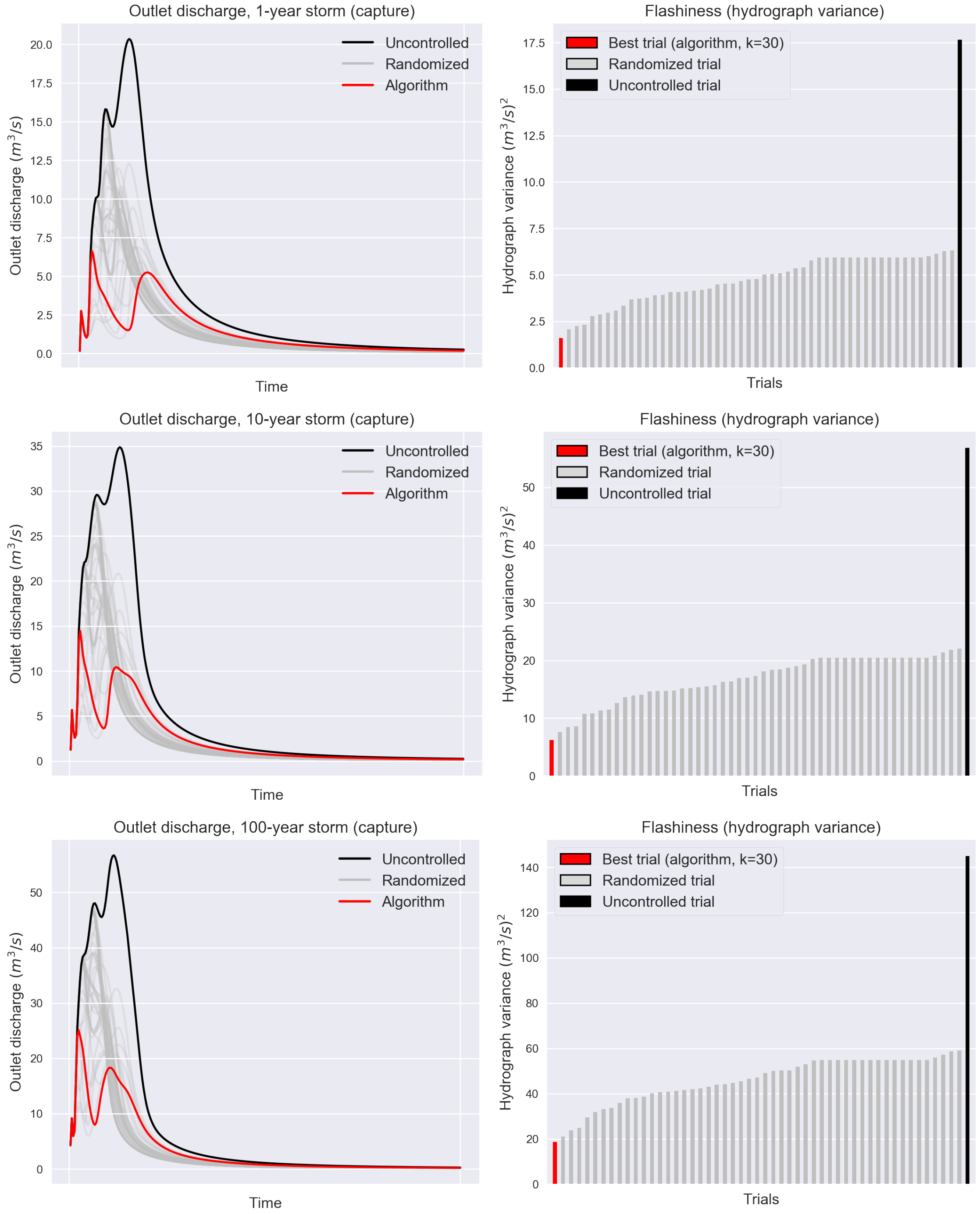


Figure S6.6. Hydrograph variance by storm size for volume capture. Left: simulated hydrographs for the uncontrolled scenario (black), randomized controller placement (gray), and the optimized controller placement (red) under 1-year, 10-year and 100-year storm events (top to bottom) for the "volume capture" scenario. Right: flashiness (as measured by the variance of the hydrograph) for each simulation.

S6.9 Peak discharge for storm events of different sizes under volume capture scenario

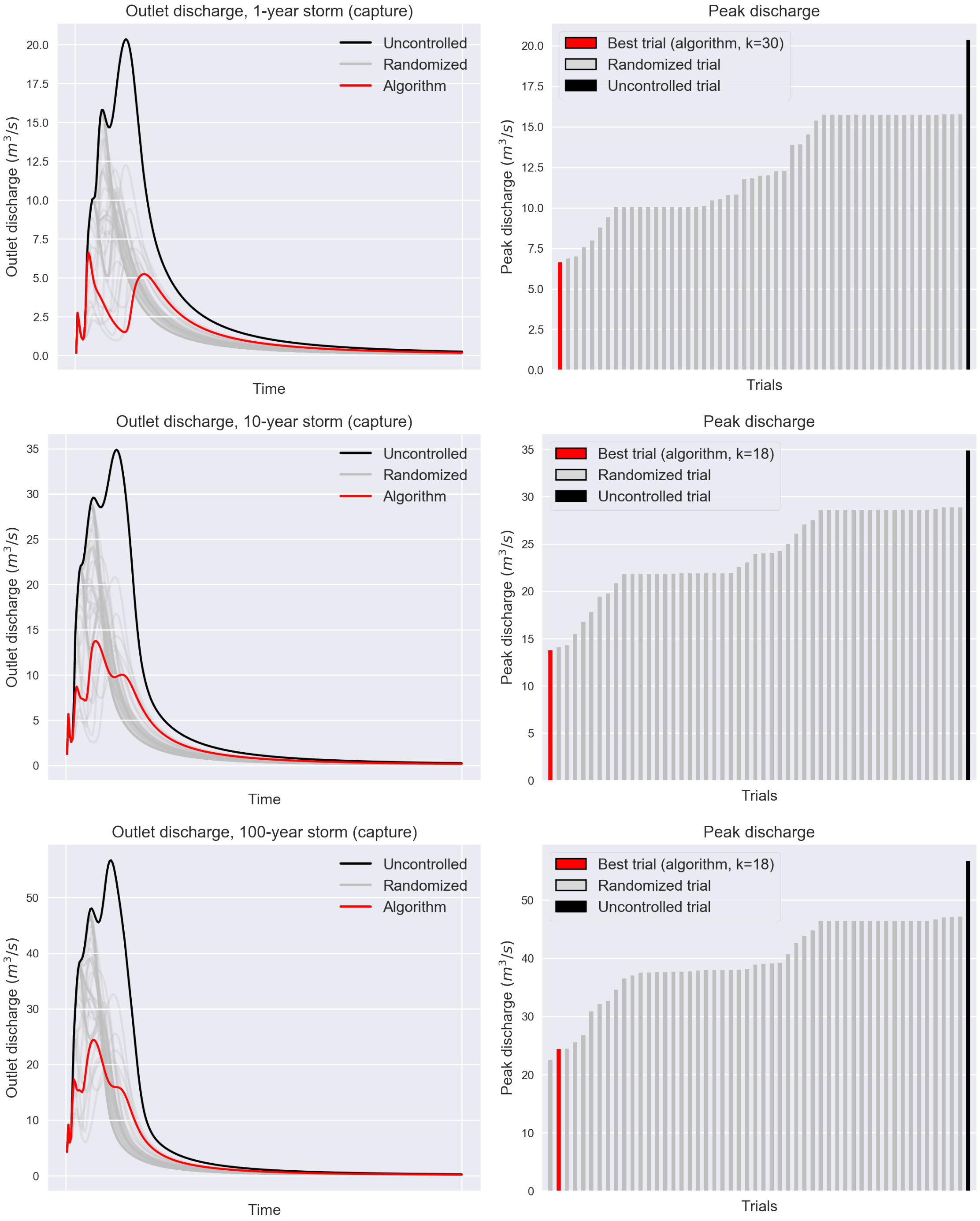


Figure S6.7. Peak discharge by storm size for volume capture. Left: simulated hydrographs for the uncontrolled scenario (black), randomized controller placement (gray), and the optimized controller placement (red) under 1-year, 10-year and 100-year storm events (top to bottom) for the "volume capture" scenario. Right: peak discharge for each simulation.

S6.10 Hydrograph variance for all simulations under volume capture

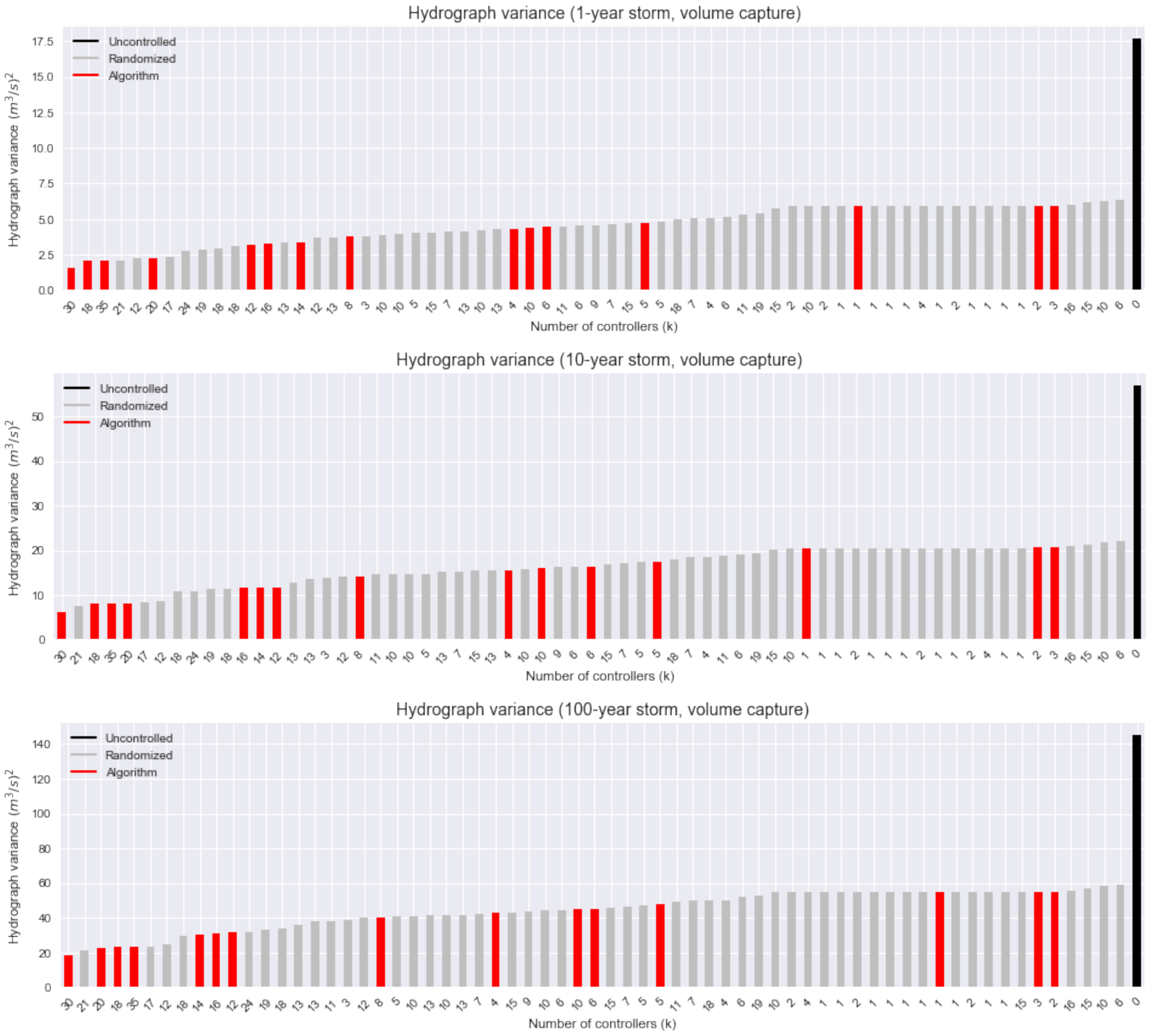


Figure S6.8. All hydrograph variances for volume capture. Hydrograph variance for 1-year, 10-year and 100-year storms under all model runs for the "volume capture" scenario.

S6.11 Peak discharge for all simulations under volume capture

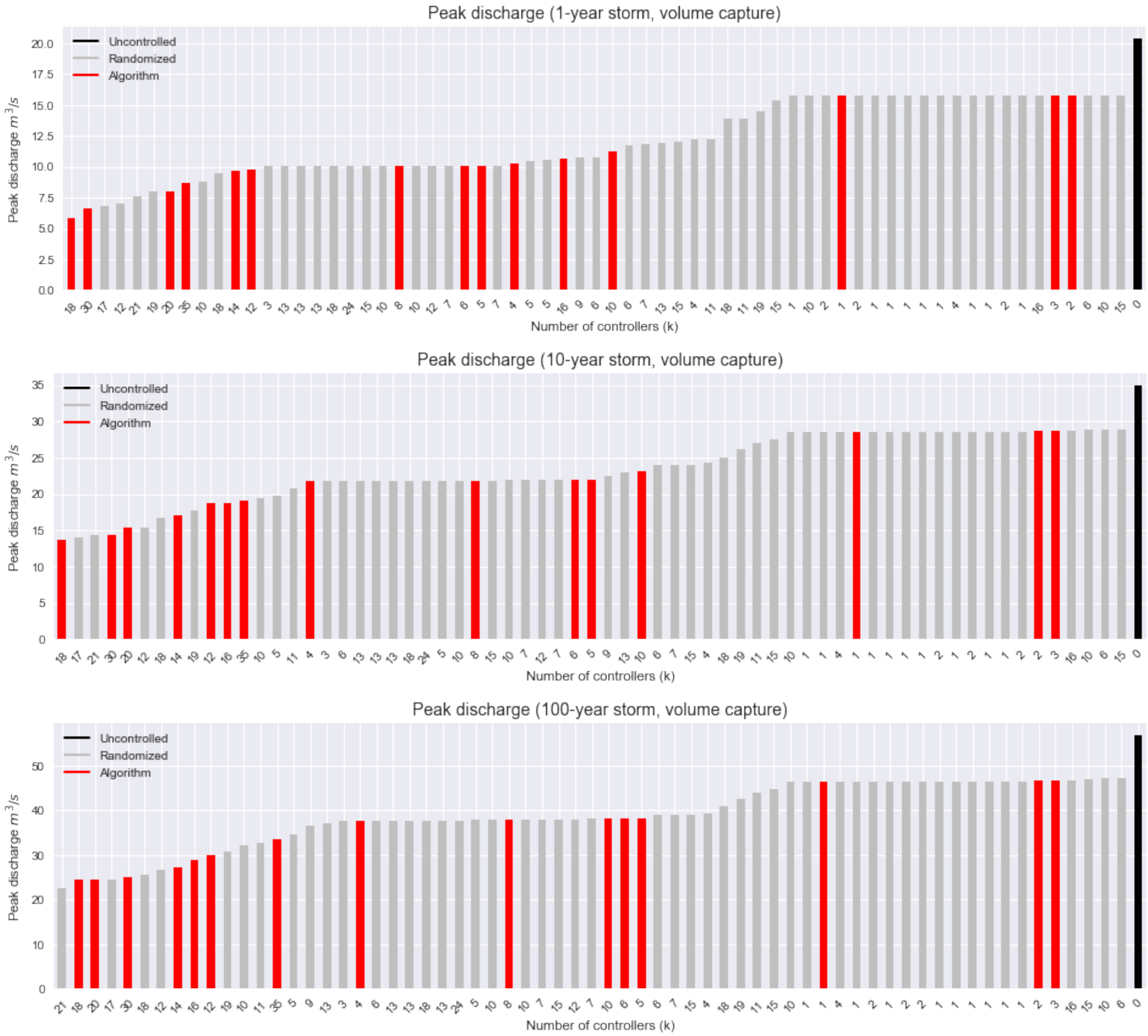


Figure S6.9. All peak discharges for volume capture. Peak discharge for 1-year, 10-year and 100-year storms under all model runs for the "volume capture" scenario.

S6.12 Hydrograph variance for storm events of different sizes under delayed release scenario

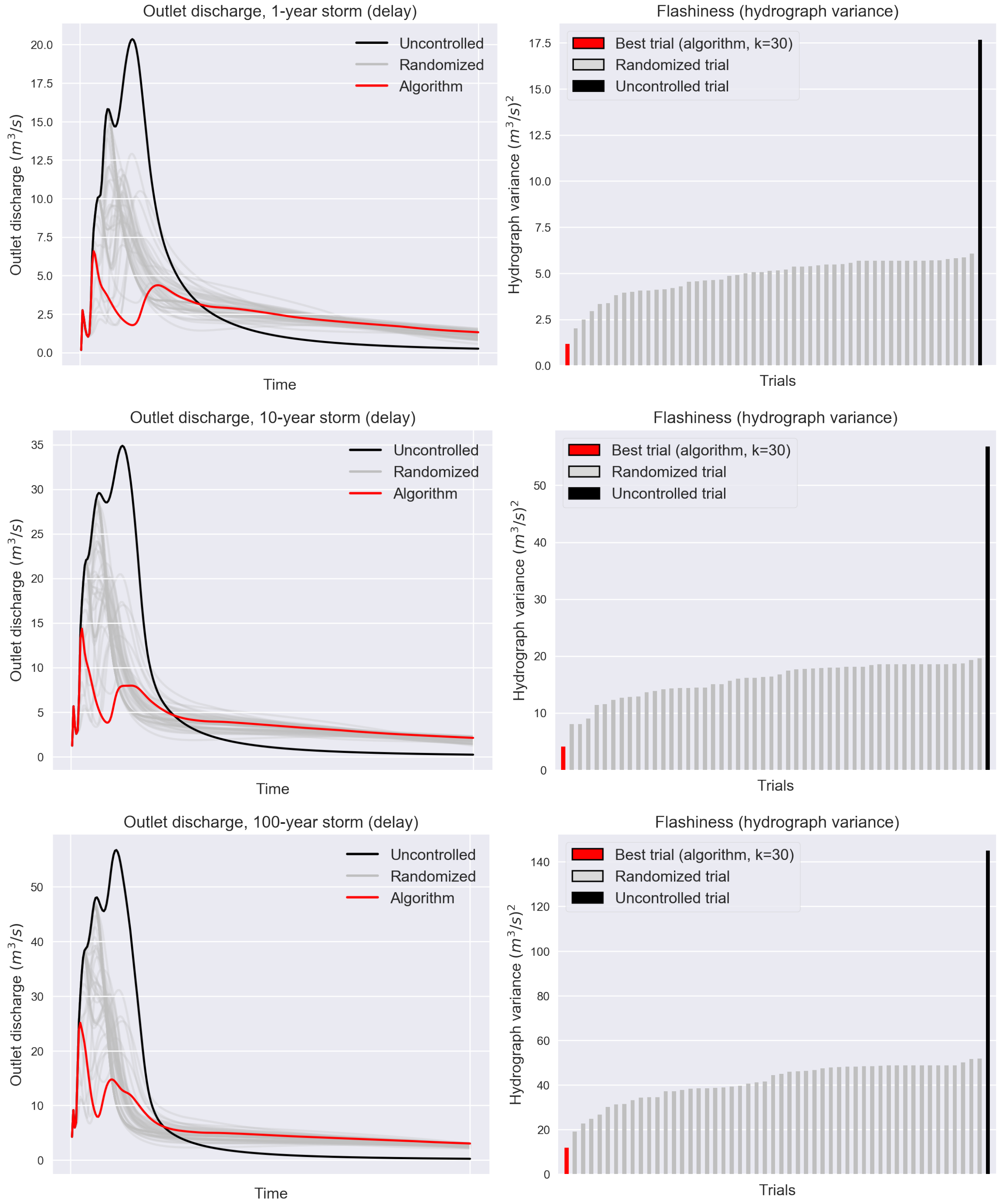


Figure S6.10. Hydrograph variance by storm size for delayed release. Left: simulated hydrographs for the uncontrolled scenario (black), randomized controller placement (gray), and the optimized controller placement (red) under 1-year, 10-year and 100-year storm events (top to bottom) under the "delayed release" scenario. Right: flashiness (as measured by the variance of the hydrograph) for each simulation.

S6.13 Peak discharge for storm events of different sizes under delayed release scenario

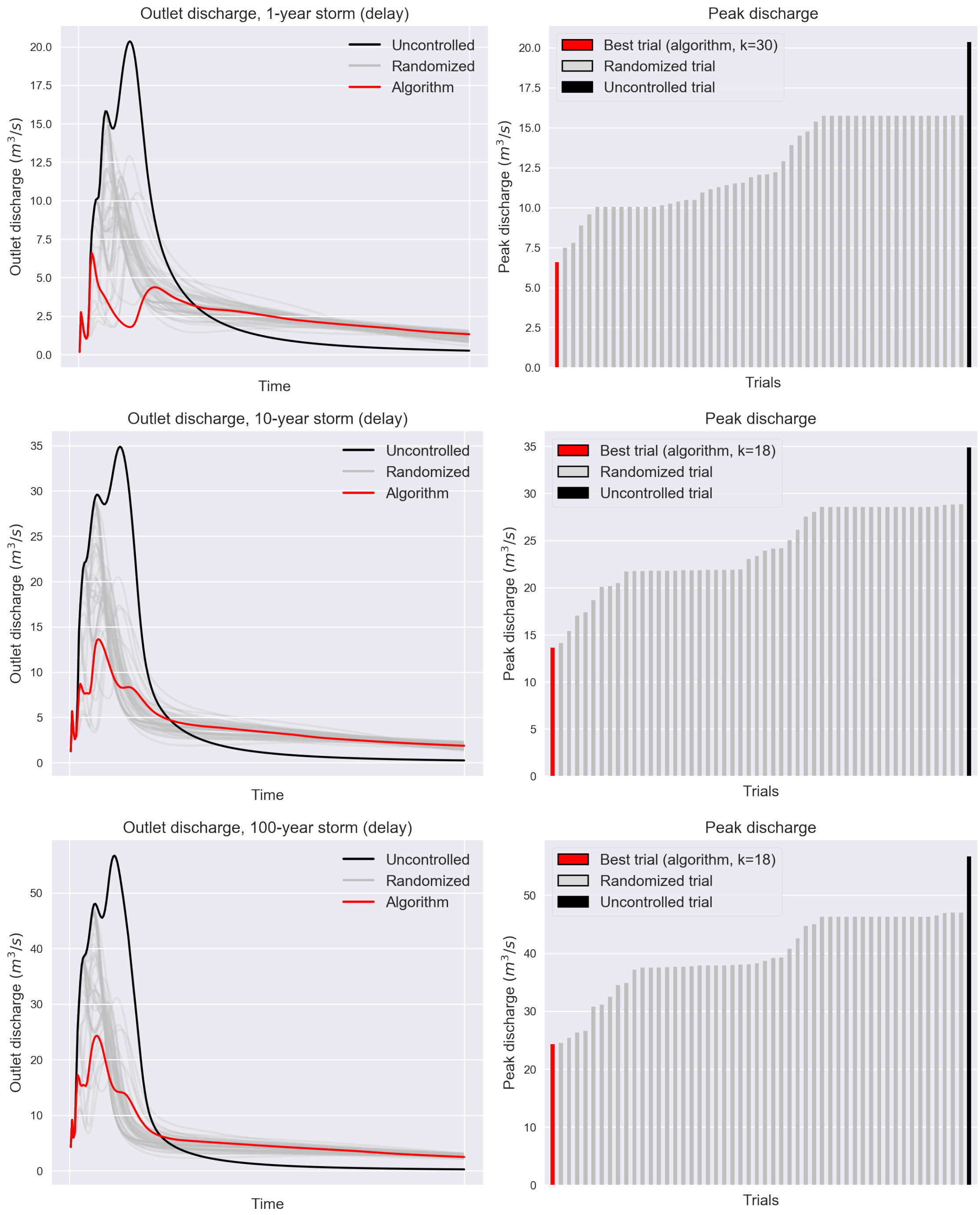


Figure S6.11. Peak discharge by storm size for delayed release. Left: simulated hydrographs for the uncontrolled scenario (black), randomized controller placement (gray), and the optimized controller placement (red) under 1-year, 10-year and 100-year storm events (top to bottom) under the "delayed release" scenario. Right: peak discharge for each simulation.

S6.14 Hydrograph variance for all simulations under delayed release

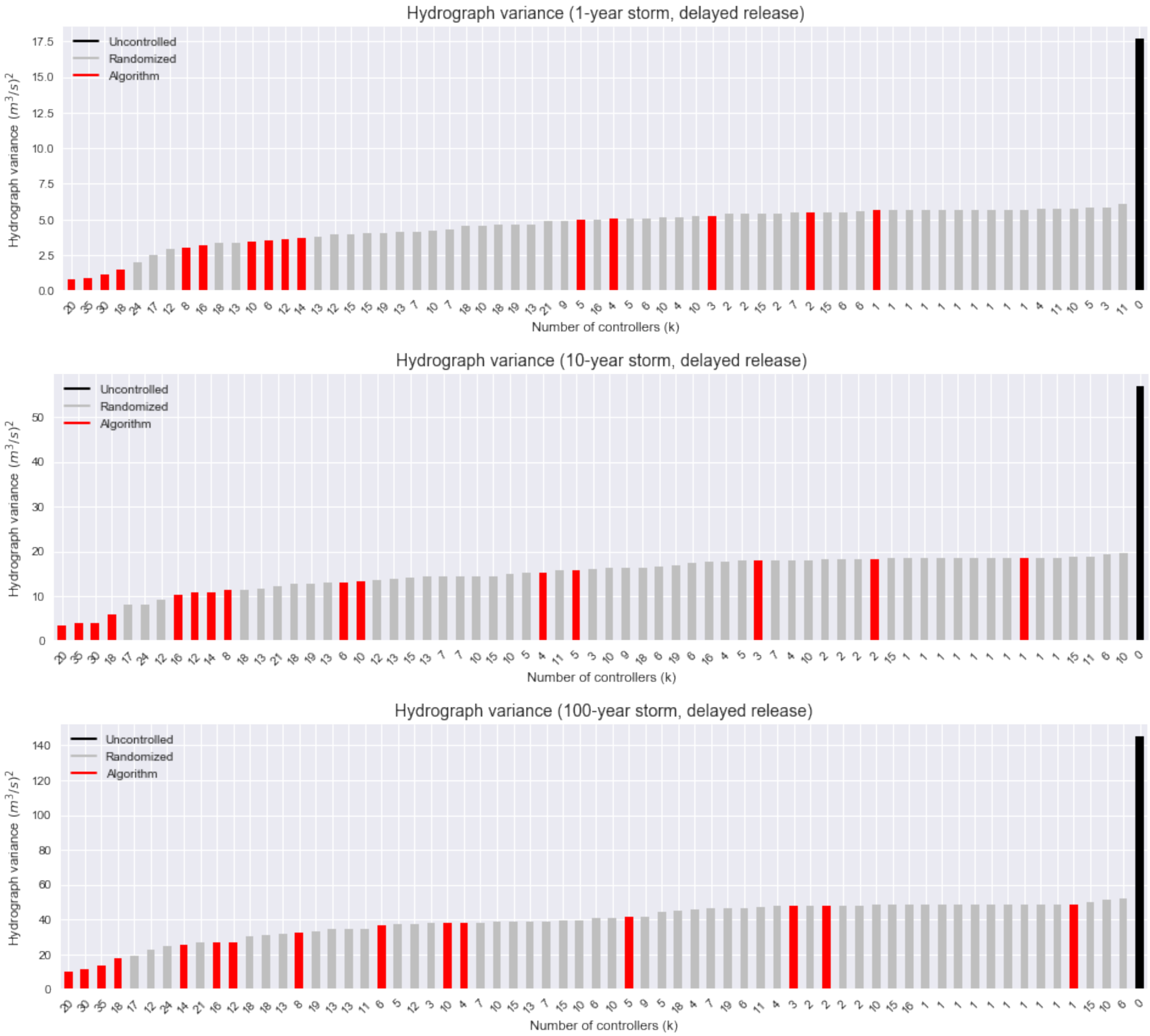


Figure S6.12. All hydrograph variances for delayed release. Hydrograph variance for 1-year, 10-year and 100-year storms under all model runs for the "delayed release" scenario.

S6.15 Peak discharge for all simulations under delayed release

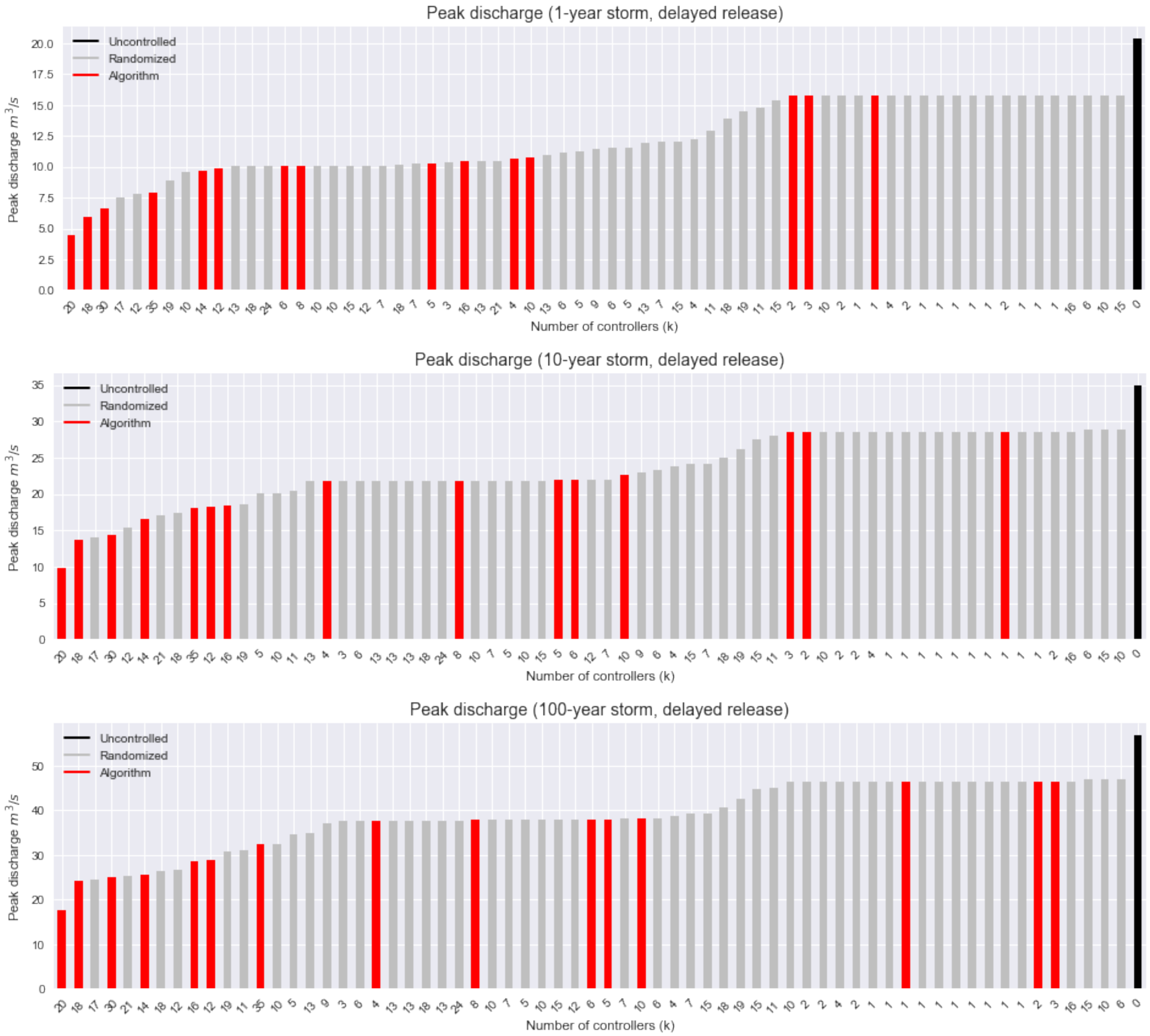


Figure S6.13. All peak discharges for delayed release. Peak discharge for 1-year, 10-year and 100-year storms under all model runs for the "delayed release" scenario.

S6.16 Performance metrics by number of controllers under volume capture

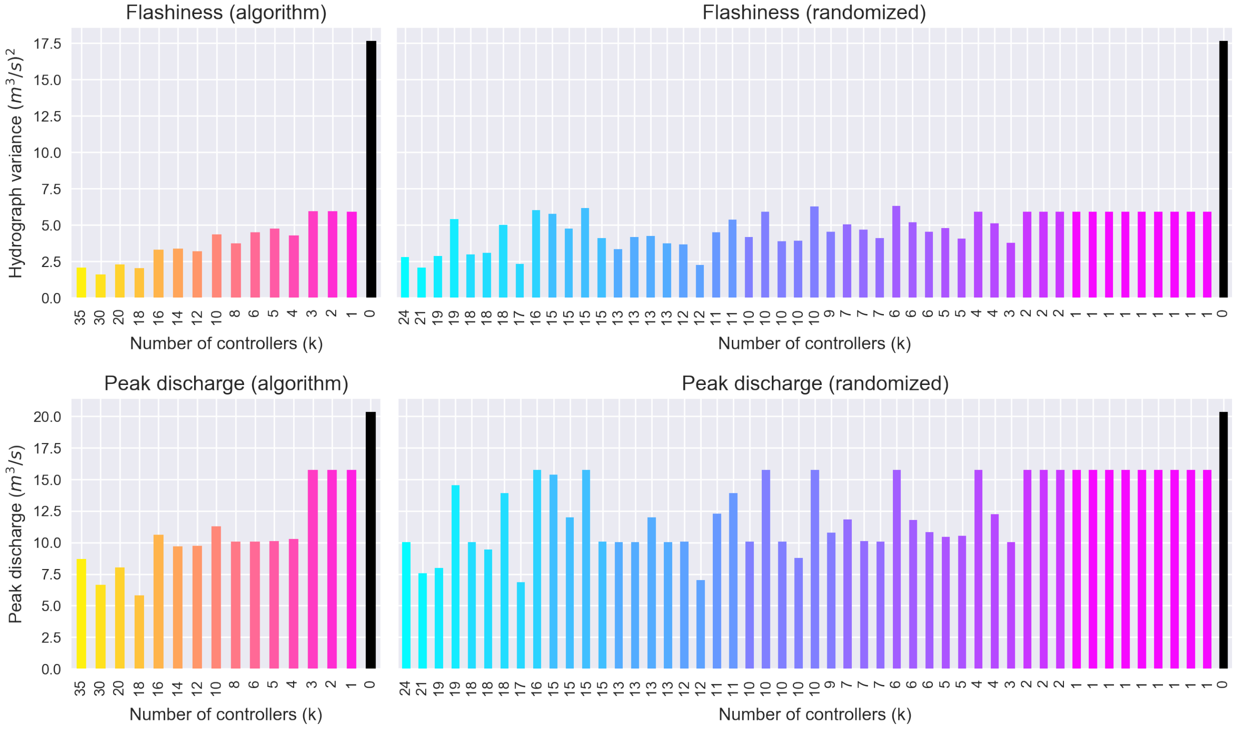


Figure S6.14. Performance metrics by number of controllers for volume capture scenario. Performance metrics are shown for both optimized (left) and randomized (right) controller placements by number of controllers used for the 1-year storm event under the "volume capture" scenario. The top panel measures performance in terms of flashiness (hydrograph variance), while the bottom panel measures performance by peak discharge. The optimized controller placements show consistently better performance metrics as the number of controllers is increased, while the randomized simulations do not.

S6.17 Performance metrics by number of controllers under delayed release

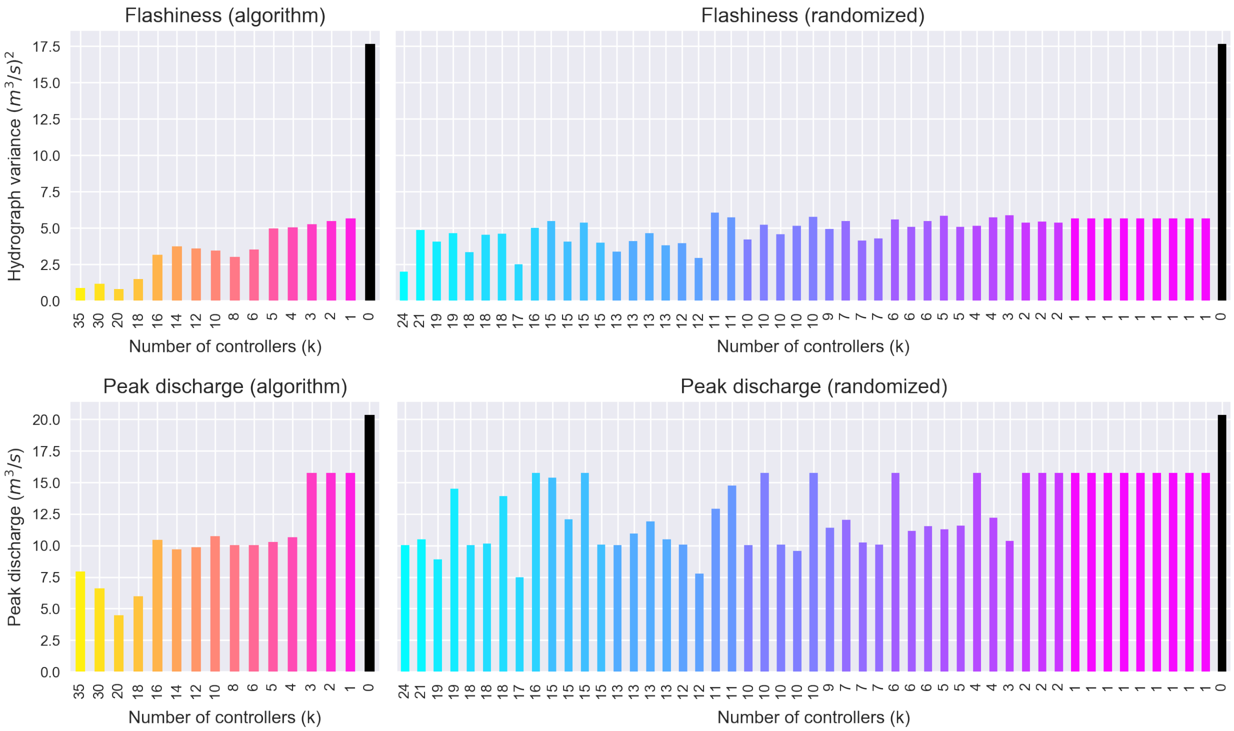


Figure S6.15. Performance metrics by number of controllers for delayed release scenario. Performance metrics are shown for both optimized (left) and randomized (right) controller placements by number of controllers used for the 1-year storm eventunder the "delayed release" scenario. The top panel measures performance in terms of flashiness (hydrograph variance), while the bottom panel measures performance by peak discharge. The optimized controller placements show consistently better performance metrics as the number of controllers is increased, while the randomized simulations do not.

Chapter 7

Conclusion

7.1 Summary of discoveries

The goal of this dissertation is to lay the groundwork for future smart urban water systems that use real-time sensing and control to improve urban flood resilience. In service of this larger goal, several fundamental discoveries are made in each chapter:

Chapter 2: I discover that combining low-power embedded devices with cloud services enables monitoring and control of urban drainage networks at previously unachievable scales. The hardware and software framework developed in this chapter will enable engineers to better understand stormwater networks and how to operate them.

Chapter 3: I discover that windshield wiper data from connected vehicles are a better indicator of the binary rainfall state (raining/not raining) than either rain gage or weather radar data. Moreover, I discover that combining windshield wiper data with weather radar data produces a new data product with greater binary detection accuracy than weather radar alone. In addition to providing emergency managers with new tools for flash flood nowcasting, this work opens the door to an entirely new field of vehicle-based environmental sensing.

Chapter 4: I discover that fusing sensor data into a hydrodynamic model improves interpolation and forecasting of hydraulic states within stormwater networks. This discovery paves the way for a new generation of *digital twin* models that fuse real-time sensor data to estimate flooding at ungaged locations, detect maintenance emergencies, and facilitate dynamic control of urban drainage networks.

Chapter 5: I discover a new numerical scheme for modeling contaminant transport in stormwater networks that is capable of maintaining numerical accuracy and stability over large time steps. This work will provide modelers with unprecedented tools for real-time modeling and state estimation of water quality in urban watersheds.

Chapter 6: I discover that the network structure of watersheds reveals important information about the optimal placement of hydraulic control structures. I then use this discovery to derive an effective heuristic algorithm for maximally attenuating peak flows by placing hydraulic control structures at strategic locations in the network. This work will provide practitioners with a powerful, general-purpose tool for siting stormwater control facilities in decentralized flood control networks.

7.2 Directions for future work

In the process of completing this thesis, many interesting research questions arose that could not be addressed within the scope of the current work. I present some of these future directions for research here:

Chapter 2: Future research should examine the problem of real-time water quality monitoring in greater detail. The sensor networks described in this chapter primarily focus on characterizing catchment hydraulics by measuring water depth. While some sensors were deployed to measure water quality indicators (such as dissolved oxygen and pH), these deployments were difficult to maintain and suffered from frequent outages and biofouling. Despite the difficulty in obtaining water quality measurements, urban water quality is a major concern for many cities and true in-situ monitoring systems would have an immense impact on stormwater management. For these reasons, future research should examine how embedded systems can be designed for long-term monitoring of water quality in urban catchments.

To increase the acceptance of smart water systems within the water resources community at large, future research should also explore new techniques for automated quality control of streaming hydrologic sensor data. With respect to the sensor network deployments described in this chapter, quality control of raw sensor data has been a persistent concern among system operators and stakeholders alike. While manual curation and filtering of data has proven effective at addressing these concerns, automated quality control of sensor data remains elusive. To date, I have not identified any algorithms that are capable of detecting and flagging all the types of sensor faults encountered in real-world hydrologic sensor data. Future work should explore the problem of real-time quality control on streaming sensor data produced by smart stormwater networks.

Chapter 3: Future work should explore how vehicle-based measurements can improve estimation of rainfall intensity. Perhaps the most significant limitation of this study is the fact that it focuses on binary detection of rainfall as opposed to rainfall intensity. The reason for this focus is twofold—first, the ground truth dataset only describes the binary rainfall state, and second, the windshield wiper measurements are not a strong predictor of rainfall intensity. However, other vehicle-based measurements have the potential to improve rainfall intensity estimates. Within this vein, one promising future research direction is the application of computer vision techniques to estimate rainfall intensity and drop size distribution from vehicle dashboard camera footage. Another promising approach is the use of onboard optical rain sensors to directly measure rainfall intensity. Future work should explore these new data sources and their potential usefulness for estimation of rainfall intensity.

Another potential limitation of this study is that there is ultimately no foolproof method for obtaining the "ground truth" rainfall state at high spatiotemporal resolutions. Given that radar and gage data are both subject to significant uncertainties, we decided to use video footage as the source of ground truth data. However, even with frame-by-frame manual review of the camera data, the potential for uncertainty still exists. For instance, droplets of water can fall on the windshield from overhanging trees. Moreover, leading vehicles can "spray" the windshields of tailing vehicles with water from the road, which can lead to some confusion over what is rain and what is not. The best way to address these limitations is to combine different sources of data together. Cross-referencing and combining radar, gage, wiper and camera data will help to eliminate uncertainties inherent to each technology in isolation, and lead to a better understanding of how localized rainfall propagates. This study presents some foundational work towards this goal, although future studies should investigate new data assimilation approaches that are better able to handle the unique uncertainties associated with each data source.

While this chapter examines the particular application of vehicle-based rainfall detection, vehicles are also capable of detecting various other environmental parameters of interest. One particularly useful application is the estimation of ambient temperature from vehicle-based thermometers. In many arid cities, urban microclimates are an important research concern with implications for ecology, public health and environmental justice. However, stationary sensor networks are often too sparse to properly characterize these microclimates. Future research should examine the potential to use vehicle-based measurements to better monitor and understand urban microclimates.

Chapter 4: Future work should investigate new methods for tuning the hydrodynamic solver's numerical scheme to ensure greater accuracy and stability. The hydraulic solver proposed in this chapter currently has a number of numerical quirks that deserve future study. First, while the numerical scheme is generally very stable, it often suffers from oscillations and spurious generation of mass. Orifices are a particular problem due to the potential for rapid flow reversals. Future work should examine methods for handling these numerical issues in greater detail. Another numerical concern is the method by which nonlinear coefficients in the continuity/momentum equations are linearized. Currently, a number of different linearization techniques are employed, and in some cases the nonlinear coefficients are approximated by simply using the value at the previous timestep. In the future, more formal methods for estimating these coefficients (e.g. Taylor expansions) should be explored.

Future work should also explore different formulations for the hydrodynamic model's boundary conditions. One of the most significant theoretical limitations of the hydraulic solver is the assumption of orifice-like flow at the boundary between superlinks and superjunctions. This assumption essentially ignores the inertial component of the momentum balance for flows entering and exiting each superjunction. While the existing formulation works well for retention basins and manholes, there are certain situations where it may perform poorly—for example, at the convergence of two tributaries where inertial forces are significant. Future research should explore how momentum flux can be incorporated into the superlink boundary conditions.

Finally, future work should explore the application of classical control algorithms to the hydrodynamic solver proposed in this chapter. The hydrodynamic solver provides a full state-space representation of stormwater system dynamics, facilitating the use of algorithms such as Linear Quadratic Regulation (LQR) and Model Predictive Control (MPC). When combined with the solver presented in this chapter, these algorithms would provide system operators with powerful tools for mitigating floods and combined sewer overflows. However, these control techniques are not explored in the current work. Future studies should explore how the hydrodynamic solver can accommodate these techniques.

Chapter 5: Future work should explore the application of state estimation and control algorithms to the water quality model developed in this chapter. Like the hydrodynamic model, the water quality model admits a state space representation that facilitates the use of powerful state estimation and control techniques. Kalman Filtering, for example, would enable better detection of contaminant releases by fusing real-time sensor data into the dynamical model. Similarly, feedback control algorithms such as LQR would enable operators to better understand how to regulate water quality within urban catchments. While these applications are not examined in the current work, they present promising directions for future studies.

Future research should also examine the effect of applying different advection schemes to the water quality model. Currently, an upwind scheme is used to handle advection; however, this scheme results in significant numerical dispersion at small timesteps. This numerical dispersion could potentially lead to misleading results when investigating highly dynamic flow regimes where small timesteps are required. Subsequent work should examine other discretization schemes for advection such as the Lax-Wendroff or Lax-Friedrichs schemes.

Finally, future work should validate the water quality model developed in this chapter using real-world data. Due to the difficulty in obtaining real-world water quality data for stormwater networks, the water quality model was validated against analytical solutions to the advection-reaction diffusion equation under simplified uniform flow conditions. However, no such analytical solutions exist for the complicated and heterogeneous flow regimes found in stormwater networks. To demonstrate the real-world effectiveness of the water quality model, future work should validate the model against tracer experiments or other real-world data.

Chapter 6: With respect to the problem of controller placement in drainage networks, the most obvious direction for future research is the complementary control-theoretic question: *what is the optimal placement of sensors?* In some preliminary work, I found that insight to this problem can be gained through analysis of the hydrodynamic system's Observability Gramian. Specifically, optimal sensor placements for different applications can be found by maximizing various scalar metrics of the Observability Gramian (such as the rank and trace). While initial results are promising, more work is needed to test and validate this sensor placement approach.

Future work should also focus on generalizing the controller placement algorithm proposed in this chapter. The algorithm incorporates a number of assumptions that may limit its general applicability. First, the algorithm assumes that the rainfall input is impulsive (i.e. with a short onset and duration). Second, the algorithm assumes uniform velocity of flow within the catchment (speed is only differentiated between hillslope and channelized segments). Finally, the algorithm assumes that the distribution of rainfall is uniform over the catchment. In many real-world applications, these assumptions may be violated. Future research should adapt the controller placement algorithm to more general conditions by revisiting some of these assumptions.

Bibliography

[1] Doocy S, Daniels A, Murray S, Kirsch TD. The Human Impact of Floods: a Historical Review of Events 1980-2009 and Systematic Literature Review. *PLoS Currents*:(2013). doi: 10.1371/currents.dis.f4deb457904936b07c09daa98ee8171a.

[2] Richardson EV, Lagasse PF. Stream Stability and Scour at Highway Bridges. *Water International* 21(3):(Sept. 1996), 108–118. doi: 10.1080/02508069608686502.

[3] Ahn JH et al. Coastal Water Quality Impact of Stormwater Runoff from an Urban Watershed in Southern California. *Environmental Science & Technology* 39(16):(Aug. 2005), 5940–5953. doi: 10.1021/es0501464.

[4] Carey RO, Wollheim WM, Mulukutla GK, Mineau MM. Characterizing Storm-Event Nitrate Fluxes in a Fifth Order Suburbanizing Watershed Using In Situ Sensors. *Environmental Science & Technology* 48(14):(July 2014), 7756–7765. doi: 10.1021/es500252j.

[5] Sercu B, Werfhorst LCVD, Murray JLS, Holden PA. Sewage Exfiltration As a Source of Storm Drain Contamination during Dry Weather in Urban Watersheds. *Environmental Science & Technology* 45(17):(Sept. 2011), 7151–7157. doi: 10.1021/es200981k.

[6] Hapuarachchi HAP, Wang QJ, Pagano TC. A review of advances in flash flood forecasting. *Hydrol. Process.* 25(18):(Mar. 2011), 2771–2784. doi: 10.1002/hyp.8040.

[7] *Clean Watersheds Needs Survey 2012*. Tech. rep. Environmental Protection Agency, 2016.

[8] Bronstert A, Niehoff D, Bürger G. Effects of climate and land-use change on storm runoff generation: present knowledge and modelling capabilities. *Hydrological processes* 16(2):(2002), 509–529.

[9] Stocker T. *Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2014.

[10] Fletcher T, Andrieu H, Hamel P. Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art. *Advances in Water Resources* 51:(Jan. 2013), 261–279. doi: 10.1016/j.advwatres.2012.09.001.

[11] Rosenberg EA et al. Precipitation extremes and the impacts of climate change on stormwater infrastructure in Washington State. *Climatic Change* 102(1-2):(Apr. 2010), 319–349. doi: 10.1007/s10584-010-9847-0.

[12] Montestruque L, Lemmon MD. Globally Coordinated Distributed Storm Water Management System. *Proceedings of the 1st ACM International Workshop on Cyber-Physical Systems for Smart Water Networks - CySWater 15*. ACM Press, 2015. doi: 10.1145/2738935.2738948.

[13] Hawley RJ, Vietz GJ. Addressing the urban stream disturbance regime. *Freshwater Science* 35(1):(Mar. 2016), 278–292. doi: 10.1086/684647.

[14] Walsh CJ et al. The urban stream syndrome: current knowledge and the search for a cure. *Journal of the North American Benthological Society* 24(3):(Sept. 2005), 706–723. doi: 10.1899/04-028.1.

[15] Wang L, Lyons J, Kanehl P, Bannerman R. Impacts of Urbanization on Stream Habitat and Fish Across Multiple Spatial Scales. *Environmental Management* 28(2):(Mar. 2001), 255–266. doi: 10.1007/s0026702409.

[16] Finkenbine JK, Atwater J, Mavinic D. Stream health after urbanization. *Journal of the American Water Resources Association* 36(5):(2000), 1149–1160.

[17] Booth DB, Jackson CR. Urbanization of aquatic systems: degradation thresholds, stormwater detection, and the limits of mitigation. *Journal of the American Water Resources Association* 33(5):(Oct. 1997), 1077–1090. doi: 10.1111/j.1752-1688.1997.tb04126.x.

[18] Wright J, Marchese D. Briefing: Continuous monitoring and adaptive control: the 'smart' storm water management solution. *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction* 170(4):(Dec. 2017), 86–89. doi: 10.1680/jsmic.17.00017.

[19] Kerkez B et al. Smarter Stormwater Systems. *Environmental Science & Technology* 50(14):(2016). PMID: 27227574, 7267–7273. doi: 10.1021/acs.est.5b05870.

[20] Gelormino MS, Ricker NL. Model-predictive control of a combined sewer system. *International Journal of Control* 59(3):(Mar. 1994), 793–816. doi: 10.1080/00207179408923105.

[21] Nooijen RR van, Kolechkina A. Speed of discrete optimization solvers for real time sewer control. *Urban Water Journal* 10(5):(Oct. 2013), 354–363. doi: 10.1080/1573062x.2013.820330.

[22] Puig V et al. Predictive optimal control of sewer networks using CORAL tool: application to Riera Blanca catchment in Barcelona. *Water Science and Technology* 60(4):(Apr. 2009), 869–878. doi: 10.2166/wst.2009.424.

[23] Andersen NK, Harremoës P, Sørensen S, Andersen HS. Monitoring and real time control in a trunk sewer. *Water Science and Technology* 36(8-9):(Oct. 1997), 337–342. doi: 10.2166/wst.1997.0689.

[24] Middleton JR, Barrett ME. Water Quality Performance of a Batch-Type Stormwater Detention Basin. *Water Environment Research* 80(2):(Feb. 2008), 172–178. doi: 10.2175/106143007x220842.

[25] Jacopin C, Lucas E, Desbordes M, Bourgogne P. Optimisation of operational management practices for the detention basins. *Water Science and Technology* 44(2-3):(July 2001), 277–285. doi: 10.2166/wst.2001.0780.

[26] Carpenter JF et al. Pollutant removal efficiency of a retrofitted stormwater detention pond. *Water Quality Research Journal* 49(2):(Dec. 2013), 124–134. doi: 10.2166/wqrjc.2013.020.

[27] Overeem A, Leijnse H, Uijlenhoet R. Country-wide rainfall maps from cellular communication networks. *Proceedings of the National Academy of Sciences* 110(8):(Feb. 2013), 2741–2745. doi: 10.1073/pnas.1217961110.

[28] Berne A, Delrieu G, Creutin JD, Obled C. Temporal and spatial resolution of rainfall measurements required for urban hydrology. *Journal of Hydrology* 299(3-4):(Dec. 2004), 166–179. doi: 10.1016/j.jhydrol.2004.08.002.

[29] Xiaoyang L, Jietai M, Yuanjing Z, Jiren L. Runoff simulation using radar and rain gauge data. *Adv. Atmos. Sci.* 20(2):(June 2003), 213–218. doi: 10.1007/s00376-003-0006-7.

[30] Yilmaz KK et al. Intercomparison of Rain Gauge, Radar, and Satellite-Based Precipitation Estimates with Emphasis on Hydrologic Forecasting. *J. Hydrometeor* 6(4):(Aug. 2005), 497–517. doi: 10.1175/jhm431.1.

[31] Sun X, Mein R, Keenan T, Elliott J. Flood estimation using radar and raingauge data. *Journal of Hydrology* 239(1-4):(Dec. 2000), 4–18. doi: 10.1016/s0022-1694(00)00350-4.

[32] Smith JA, Krajewski WF. A modeling study of rainfall rate-reflectivity relationships. *Water Resources Research* 29(8):(Aug. 1993), 2505–2514. doi: 10.1029/93wr00962.

[33] Smith JA, Seo DJ, Baeck ML, Hudlow MD. An Intercomparison Study of NEXRAD Precipitation Estimates. *Water Resources Research* 32(7):(July 1996), 2035–2045. doi: 10.1029/96wr00270.

[34] Morin E et al. Estimating rainfall intensities from weather radar data: the scale-dependency problem. *Journal of Hydrometeorology* 4(5):(Oct. 2003), 782–797. doi: 10.1175/1525-7541(2003)004<0782:ERIFWR>2.0.CO;2.

[35] Islam T, Rico-Ramirez MA, Han D, Srivastava PK. Artificial intelligence techniques for clutter identification with polarimetric radar signatures. *Atmospheric Research* 109-110:(June 2012), 95–113. doi: 10.1016/j.atmosres.2012.02.007.

[36] Hill D et al. Sensing and Cyberinfrastructure for Smarter Water Management: The Promise and Challenge of Ubiquity. *J. Water Resour. Plann. Manage.* 140(7):(July 2014), 01814002. doi: 10.1061/(asce)wr.1943-5452.0000449.

[37] Hill DJ. Assimilation of weather radar and binary ubiquitous sensor measurements for quantitative precipitation estimation. *Journal of Hydroinformatics* 17(4):(July 2015), 598. doi: 10.2166/hydro.2015.072.

[38] Haberlandt U, Sester M. Areal rainfall estimation using moving cars as rain gauges – a modelling study. *Hydrol. Earth Syst. Sci.* 14(7):(July 2010), 1139–1151. doi: 10.5194/hess-14-1139-2010.

[39] Karmous-Edwards G. Technology Perspectives: Digital Twin Technology Becoming a Reality for Water Utilities. *Water Finance & Management*:(June 2019).

[40] Ji Z. General Hydrodynamic Model for Sewer/Channel Network Systems. *Journal of Hydraulic Engineering* 124(3):(Mar. 1998), 307–315. doi: 10.1061/(asce)0733-9429(1998)124:3(307).

[41] Martin JL, McCutcheon SC, Schottman RW. *Hydrodynamics and Transport for Water Quality Modeling*. CRC Press, May 2018. doi: 10.1201/9780203751510.

[42] Ewing RE, Wang H. A summary of numerical methods for time-dependent advection-dominated partial differential equations. *Partial Differential Equations*. Elsevier, 2001, pp. 423–445. doi: 10.1016/b978-0-444-50616-0.50018-x.

[43] Rossman LA, Huber WC. *Storm Water Management Model Reference Manual Volume III - Water Quality*. National Risk Management Laboratory, Office of Research and Development, 2016.

[44] Wong BP, Kerkez B. Real-time control of urban headwater catchments through linear feedback: performance, analysis and site selection. *Water Resources Research.*(Sept. 2018). doi: 10.1029/2018wr022657.

[45] Ayalew TB, Krajewski WF, Mantilla R. Insights into Expected Changes in Regulated Flood Frequencies due to the Spatial Configuration of Flood Retention Ponds. *Journal of Hydrologic Engineering* 20(10):(Oct. 2015), 04015010. doi: 10.1061/(asce)he.1943-5584.0001173.

[46] Schilling J, Logan J. Greening the rust belt: A green infrastructure model for right sizing America's shrinking cities. *Journal of the American Planning Association* 74(4):(2008), 451–466.

[47] Yao L, Chen L, Wei W, Sun R. Potential reduction in urban runoff by green spaces in Beijing: A scenario analysis. *Urban Forestry & Urban Greening* 14(2):(2015), 300–308. issn: 1618-8667. doi: 10.1016/j.ufug.2015.02.014.

[48] Zellner M, Massey D, Minor E, Gonzalez-Meler M. Exploring the effects of green infrastructure placement on neighborhood-level flooding via spatially explicit simulations. *Computers, Environment and Urban Systems* 59:(2016), 116–128. issn: 0198-9715. doi: 10.1016/j.compenvurbsys.2016.04.008.

[49] Riasi MS, Yeghiazarian L. Controllability of Surface Water Networks. *Water Resources Research* 53(12):(Dec. 2017), 10450–10464. doi: 10.1002/2017wr020861.

[50] Caragliu A, Bo CD, Nijkamp P. Smart Cities in Europe. *Journal of Urban Technology* 18(2):(Apr. 2011), 65–82. doi: 10.1080/10630732.2011.601117.

[51] Dimitrakopoulos G, Demestichas P. Intelligent Transportation Systems. *IEEE Vehicular Technology Magazine* 5(1):(Mar. 2010), 77–84. doi: 10.1109/mvt.2009.935537.

[52] Zanella A et al. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1(1):(Feb. 2014), 22–32. doi: 10.1109/jiot.2014.2306328.

[53] Lynch JP. Design of a wireless active sensing unit for localized structural health monitoring. *Structural Control and Health Monitoring* 12(3-4):(2005), 405–423. doi: 10.1002/stc.77.

[54] Mays L. *Water Resources Engineering*. John Wiley & Sons, 2010. isbn: 9780470460641.

[55] Vörösmarty CJ et al. Global threats to human water security and river biodiversity. *Nature* 467(7315):(Sept. 2010), 555–561. doi: 10.1038/nature09440.

[56] *2013 Report Card for America's Infrastructure*. American Society of Civil Engineers, 2013. doi: 10.1061/9780784478837.

[57] Emerson CH, Welty C, Traver RG. Watershed-Scale Evaluation of a System of Storm Water Detention Basins. *Journal of Hydrologic Engineering* 10(3):(May 2005), 237–242. doi: 10.1061/(asce)1084-0699(2005)10:3(237).

[58] *The Importance of Operation and Maintenance for the Long-Term Success of Green Infrastructure: A Review of Green Infrastructure O&M Practices in ARRA Clean Water State Revolving Fund Projects*. Tech. rep. Environmental Protection Agency, 2013.

[59] Wong BP, Kerkez B. Real-time environmental sensor data: An application to water quality using web services. *Environmental Modelling & Software* 84:(Oct. 2016), 505–517. doi: 10.1016/j.envsoft.2016.07.020.

[60] Mullapudi A, Wong BP, Kerkez B. Emerging investigators series: building a theory for smart stormwater systems. *Environ. Sci.: Water Res. Technol.* 3(1):(2017), 66–77. doi: 10.1039/c6ew00211k.

[61] Mays LW et al. *Water distribution systems handbook*. Vol. 17. McGraw-Hill New York, 2000.

[62] Powell R, Hindi K. *Computing and Control for the Water Industry*. Computing and Control for the Water Industry. Research Studies Press, 1999. isbn: 9780863802300.

[63] Association A. *Instrumentation and Control, 3rd Ed. (M2):* Manual of Water Supply Practices. American Water Works Association, 2001. isbn: 9781583211250.

[64] Igure VM, Laughter SA, Williams RD. Security issues in {SCADA} networks. *Computers & Security* 25(7):(2006), 498–506. issn: 0167-4048. doi: https://doi.org/10.1016/j.cose.2006.03.001.

[65] Mays L. *Water supply systems security*. McGraw-Hill professional engineering. McGraw-Hill, 2004. isbn: 9780071425315.

[66] Cerpa A et al. Habitat monitoring. *Workshop on Data communication in Latin America and the Caribbean - SIGCOMM LA 01*. ACM Press, 2001. doi: 10.1145/371626.371720.

[67] Hefeeda M, Bagheri M. Wireless Sensor Networks for Early Detection of Forest Fires. *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*. IEEE, Oct. 2007. doi: 10.1109/mobhoc.2007.4428702.

[68] Soliman H, Sudan K, Mishra A. A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks. *2010 IEEE Sensors*. IEEE, Nov. 2010. doi: 10.1109/icsens.2010.5690033.

[69] Kim Y, Evans RG, Iversen WM. Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network. *IEEE Transactions on Instrumentation and Measurement* 57(7):(July 2008), 1379–1387. issn: 0018-9456. doi: 10.1109/TIM.2008.917198.

[70] Martinez K, Hart JK, Ong R. Environmental sensor networks. *Computer* 37(8):(Aug. 2004), 50–56. issn: 0018-9162. doi: 10.1109/MC.2004.91.

[71] Bonnet P, Gehrke J, Seshadri P. Querying the physical world. *IEEE Personal Communications* 7(5):(2000), 10–15. doi: 10.1109/98.878531.

[72] Imielinski T, Goel S. DataSpace: querying and monitoring deeply networked collections in physical space. *IEEE Personal Communications* 7(5):(2000), 4–9. doi: 10.1109/98.878530.

[73] Chen N, Wang K, Xiao C, Gong J. A heterogeneous sensor web node meta-model for the management of a flood monitoring system. *Environmental Modelling & Software* 54:(Apr. 2014), 222–237. doi: 10.1016/j.envsoft.2014.01.014.

[74] Hughes D et al. An experiment with reflective middleware to support grid-based flood monitoring. *Concurrency and Computation: Practice and Experience* 20(11):(2008), 1303–1316. issn: 1532-0634. doi: 10.1002/cpe.1279.

[75] Smith PJ et al. Towards the provision of site specific flood warnings using wireless sensor networks. *Meteorological Applications* 16(1):(2009), 57–64. issn: 1469-8080. doi: 10.1002/met.130.

[76] Basha EA, Ravela S, Rus D. Model-based monitoring for early warning flood detection. *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys 08*. ACM Press, 2008. doi: 10.1145/1460412.1460442.

[77] Demir I, Krajewski WF. Towards an integrated Flood Information System: Centralized data access, analysis, and visualization. *Environmental Modelling & Software* 50:(Dec. 2013), 77–84. doi: 10.1016/j.envsoft.2013.08.009.

[78] Marin-Perez R, García-Pintado J, Gómez AS. A Real-Time Measurement System for Long-Life Flood Monitoring and Warning Applications. *Sensors* 12(12):(Mar. 2012), 4213–4236. doi: 10.3390/s120404213.

[79] See C et al. A Low Power Wireless Sensor Network for Gully Pot Monitoring in Urban Catchments. *IEEE Sensors Journal*:(2011). doi: 10.1109/jsen.2011.2174354.

[80] Quigley M. Advanced Rainwater Harvesting for Urban Stormwater Management. *EWRI Currents* 17(3):(2015).

[81] *Elastic Compute Cloud (EC2): Cloud Server & Hosting, AWS*. 2017. url: https://aws.amazon.com/ec2/.

[82] *Microsoft Azure: Cloud Computing Platform & Services*. 2017. url: https://azure.microsoft.com.

[83] *InfluxDB: Scalable datastore for metrics, events, and real-time analytics*. 2017. url: https://www.influxdata.com/.

[84] *Grafana – the open platform for analytics and monitoring*. 2017. url: https://grafana.com.

[85] *Twitter. It's what's happening*. 2017. url: https://twitter.com.

[86] *Slack: Where work happens*. 2017. url: https://slack.com.

[87] Wong BP, Kerkez B. Adaptive measurements of urban runoff quality. *Water Resources Research* 52(11):(2016), 8986–9000. issn: 1944-7973. doi: 10.1002/2015WR018013.

[88] Riaño-Briceño G et al. MatSWMM – An open-source toolbox for designing real-time control of urban drainage systems. *Environmental Modelling & Software* 83:(Sept. 2016), 143–154. doi: 10.1016/j.envsoft.2016.05.009.

[89] *pySWMM: Python Wrappers for SWMM*. 2017. url: https://github.com/OpenWaterAnalytics/pyswmm.

[90] Kerkez B et al. Cloud Hosted Real-time Data Services for the Geosciences (CHORDS). *Geoscience Data Journal* 3(1):(June 2016), 4–8. doi: 10.1002/gdj3.36.

[91] Sharif HO, Jackson TL, Hossain MM, Zane D. Analysis of Flood Fatalities in Texas. *Natural Hazards Review* 16(1):(2014). doi: 10.1061/(ASCE)NH.1527-6996.0000145.

[92] Lee L. Do you live in flash flood alley? Experts explain the central Texas phenomenon and what residents should know. *txH2O*:(Fall 2016 2016).

[93] Smith JA et al. Radar rainfall estimation for flash flood forecasting in small urban watersheds. *Advances in Water Resources* 30(10):(Oct. 2007), 2087–2097. doi: 10.1016/j.advwatres.2006.09.007.

[94] *F6 Climate Data*. 2017. url: https://www.weather.gov/fwd/f6.

[95] *Texas Water Dashboard*. 2017. url: https://txpub.usgs.gov/txwaterdashboard.

[96] Sheehan H. Personal communication. July 5, 2017.

[97] Crum TD, Alberty RL. The WSR-88D and the WSR-88D Operational Support Facility. *Bulletin of the American Meteorological Society* 74(9):(Sept. 1993), 1669–1687. doi: 10.1175/1520-0477(1993)074<1669:twatwo>2.0.co;2.

[98] Jiang D, Delgrossi L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. *VTC Spring 2008 - IEEE Vehicular Technology Conference*. May 2008, pp. 2036–2040. doi: 10.1109/VETECS.2008.458.

[99] Lawson R, Riggs E, Weiker D, Doubek J. *Total Suspended Solids Reduction and Implementation Plan for Malletts Creek: October 2011 - September 2016*. Tech. rep. Huron River Watershed Council, 2016.

[100] *Dark Sky API*. 2017. url: https://darksky.net/dev.

[101] Gaborit E et al. Improving the performance of stormwater detention basins by real-time control using rainfall forecasts. *Urban Water Journal* 10(4):(Aug. 2013), 230–246. doi: 10.1080/1573062x.2012.726229.

[102] Bartos M, Wong B, Kerkez B. Open storm: a complete framework for sensing and control of urban watersheds. *Environmental Science: Water Research & Technology* 4(3):(2018), 346–358. doi: 10.1039/c7ew00374a.

[103] Cheng L, AghaKouchak A. Nonstationary Precipitation Intensity-Duration-Frequency Curves for Infrastructure Design in a Changing Climate. *Scientific Reports* 4(1):(Nov. 2014). doi: 10.1038/srep07093.

[104] Grimes D, Pardo-Igúzquiza E, Bonifacio R. Optimal areal rainfall estimation using raingauges and satellite data. *Journal of Hydrology* 222(1-4):(Sept. 1999), 93–108. doi: 10.1016/s0022-1694(99)00092-x.

[105] Winchell M, Gupta HV, Sorooshian S. On the simulation of infiltration- and saturation-excess runoff using radar-based rainfall estimates: Effects of algorithm uncertainty and pixel aggregation. *Water Resources Research* 34(10):(Oct. 1998), 2655–2670. doi: 10.1029/98wr02009.

[106] Wong B. Real-time measurement and control of urban stormwater systems. PhD thesis. University of Michigan, 2017.

[107] Weather Underground. Weather Underground Personal Weather Stations (City of Ann Arbor). 2014.

[108] NOAA National Weather Service (NWS) Radar Operations Center. NOAA Next Generation Radar (NEXRAD) Level 3 Products (Instantaneous Precipitation Rate). 1992. doi: 10.7289/V5W9574V.

[109] University of Michigan Transportation Research Institute. Safety Pilot Model Deployment / Ann Arbor Connected Vehicle Test Environment Data. 2014.

[110] Park H, Liu J, Johnson-Roberson M, Vasudevan R. Robust Environmental Mapping by Mobile Sensor Networks. *IEEE International Conference on Robotics and Automation*. URL: https://arxiv.org/abs/1711.07510. 2018.

[111] Berzuini C, Best NG, Gilks WR, Larizza C. Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association* 92(440):(1997), 1403–1412.

[112] Bartos M et al. Windshield wipers on connected vehicles produce high-accuracy rainfall maps. *Scientific Reports* 9(1):(Jan. 2019). doi: 10.1038/s41598-018-36282-7.

[113] Rosenberg EA et al. Precipitation extremes and the impacts of climate change on stormwater infrastructure in Washington State. *Climatic Change* 102(1-2):(Apr. 2010), 319–349. doi: 10.1007/s10584-010-9847-0.

[114] Blecken GT et al. Stormwater control measure (SCM) maintenance considerations to ensure designed functionality. *Urban Water Journal* 14(3):(Nov. 2015), 278–290. doi: 10.1080/1573062x.2015.1111913.

[115] Criss RE, Shock EL. Flood Enhancement through Flood Control. *Geology* 29(10):(2001), 875. issn: 0091-7613. doi: 10.1130/0091-7613(2001)029<0875:FETFC>2.0.CO;2.

[116] Huber W, Dickinson R, Rossman L. EPA Storm Water Management Model, SWMM5. *Watershed Models*. CRC Press, Sept. 2005, pp. 338–359. doi: 10.1201/9781420037432.ch14.

[117] Todini E. A Bayesian technique for conditioning radar precipitation estimates to rain-gauge measurements. *Hydrology and Earth System Sciences* 5(2):(2001), 187–199. doi: 10.5194/hess-5-187-2001.

[118] Simon D. *Optimal State Estimation*. John Wiley & Sons, Inc., Jan. 2006. doi: 10.1002/0470045345.

[119] Liu Y et al. Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities. *Hydrology and Earth System Sciences* 16(10):(Oct. 2012), 3863–3887. doi: 10.5194/hess-16-3863-2012.

[120] Seo DJ, Koren V, Cajina N. Real-Time Variational Assimilation of Hydrologic and Hydrometeorological Data into Operational Hydrologic Forecasting. *Journal of Hydrometeorology* 4(3):(2003), 627–641. doi: 10.1175/1525-7541(2003)004<0627:RVAOHA>2.0.CO;2.

[121] Thirel G et al. A past discharges assimilation system for ensemble streamflow forecasts over France – Part 1: Description and validation of the assimilation system. *Hydrology and Earth System Sciences* 14(8):(Aug. 2010), 1623–1637. doi: 10.5194/hess-14-1623-2010.

[122] Weerts AH et al. Application of generic data assimilation tools (DATools) for flood forecasting purposes. *Computers & Geosciences* 36(4):(Apr. 2010), 453–463. doi: 10.1016/j.cageo.2009.07.009.

[123] Dechant C, Moradkhani H. Radiance data assimilation for operational snow and streamflow forecasting. *Advances in Water Resources* 34(3):(Mar. 2011), 351–364. doi: 10.1016/j.advwatres.2010.12.009.

[124] Leeuwen PJ van. Particle Filtering in Geophysical Systems. *Monthly Weather Review* 137(12):(Dec. 2009), 4089–4114. doi: 10.1175/2009mwr2835.1.

[125] Pauwels VRN, Lannoy GJMD. Ensemble-based assimilation of discharge into rainfall-runoff models: A comparison of approaches to mapping observational information to state space. *Water Resources Research* 45(8):(Aug. 2009). doi: 10.1029/2008wr007590.

[126] Schwanenberg D, Breukelen A van, Hummel S. Data assimilation for supporting optimum control in large-scale river networks. *2011 International Conference on Networking, Sensing and Control*. IEEE, Apr. 2011. doi: 10.1109/icnsc.2011.5874881.

[127] Dotto CB et al. Comparison of different uncertainty techniques in urban stormwater quantity and quality modelling. *Water Research* 46(8):(May 2012), 2545–2558. doi: 10.1016/j.watres.2012.02.009.

[128] Schaefli B, Talamba DB, Musy A. Quantifying hydrological modeling errors through a mixture of normal distributions. *Journal of Hydrology* 332(3-4):(Jan. 2007), 303–315. doi: 10.1016/j.jhydrol.2006.07.005.

[129] Schoups G, Vrugt JA. A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-Gaussian errors. *Water Resources Research* 46(10):(Oct. 2010). doi: 10.1029/2009wr008933.

[130] Thyer M et al. Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using Bayesian total error analysis. *Water Resources Research* 45(12):(Apr. 2009). doi: 10.1029/2008wr006825.

[131] Renard B et al. Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors. *Water Resources Research* 46(5):(May 2010). doi: 10.1029/2009wr008328.

[132] Hansen L, Borup M, Møller A, Mikkelsen P. Flow Forecasting using Deterministic Updating of Water Levels in Distributed Hydrodynamic Urban Drainage Models. *Water* 6(8):(July 2014), 2195–2211. doi: 10.3390/w6082195.

[133] Borup M. Real Time Updating in Distributed Urban Rainfall Runoff Modelling:(2014).

[134] Hutton C, Kapelan Z, Vamvakeridou-Lyroudia L, Savić D. Real-time Data Assimilation in Urban Rainfall-runoff Models. *Procedia Engineering* 70:(2014), 843–852. doi: 10.1016/j.proeng.2014.02.092.

[135] Breinholt A et al. Grey-box modelling of flow in sewer systems with state-dependent diffusion. *Environmetrics* 22(8):(Oct. 2011), 946–961. doi: 10.1002/env.1135.

[136] Breinholt A, Møller JK, Madsen H, Mikkelsen PS. A formal statistical approach to representing uncertainty in rainfall–runoff modelling with focus on residual analysis and probabilistic output evaluation – Distinguishing simulation and prediction. *Journal of Hydrology* 472-473:(Nov. 2012), 36–52. doi: 10.1016/j.jhydrol.2012.09.014.

[137] Zahner JA. Influence of Storm Sewers, Drainage Density, and Soil Moisture on Runoff from an Urbanizing Catchment. PhD thesis. University of Connecticut, 2004.

[138] Freni G, Maglionico M, Federico VD. *State of the art in Urban Drainage Modelling*. Tech. rep. EVK1-CT-2001-00167. University of Bologna, 2003.

[139] Cunge J. Practical aspects of computational river hydraulics. *Pitman Publishing Ltd. London,(17 CUN), 1980, 420*:(1980).

[140] Kalman RE. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82(1):(Mar. 1960), 35–45. doi: 10.1115/1.3662552.

[141] Skliar M, Ramirez WF. Implicit Kalman filtering. *International Journal of Control* 66(3):(Jan. 1997), 393–412. doi: 10.1080/002071797224649.

[142] Söderlind G. Digital filters in adaptive time-stepping. *ACM Transactions on Mathematical Software* 29(1):(Mar. 2003), 1–26. doi: 10.1145/641876.641877.

[143] Green WH, Ampt GA. Studies on Soil Phyics. *The Journal of Agricultural Science* 4(1):(May 1911), 1–24. doi: 10.1017/s0021859600001441.

[144] Rossum G van. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam:(1995).

[145] Lam SK, Pitrou A, Seibert S. Numba: A LLVM-based Python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM 2015*. ACM Press, 2015. doi: 10.1145/2833157.2833162.

[146] Cuthill E, McKee J. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 1969 24th national conference on -*. ACM Press, 1969. doi: 10.1145/800195.805928.

[147] Rimer SP, Mullapudi A, Troutman SC, Kerkez B. A benchmarking framework for control and optimization of smart stormwater networks. *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. ACM, Apr. 2019. doi: 10.1145/3302509.3313336.

[148] Habibi H et al. High-resolution hydrologic forecasting for very large urban areas. *Journal of Hydroinformatics* 21(3):(Feb. 2019), 441–454. doi: 10.2166/hydro.2019.100.

[149] Ni K et al. Sensor network data fault types. *ACM Transactions on Sensor Networks* 5(3):(May 2009), 1–29. doi: 10.1145/1525856.1525863.

[150] Agency USEP. Report to Congress on Impacts and Control of Combined Sewer Overflows and Sanitary Sewer Overflows:(2004).

[151] Calhoun D, LeVeque RJ. A Cartesian Grid Finite-Volume Method for the Advection-Diffusion Equation in Irregular Geometries. *Journal of Computational Physics* 157(1):(Jan. 2000), 143–180. doi: 10.1006/jcph.1999.6369.

[152] Tayebi A, Shekari Y, Heydari M. A meshless method for solving two-dimensional variable-order time fractional advection–diffusion equation. *Journal of Computational Physics* 340:(July 2017), 655–669. doi: 10.1016/j.jcp.2017.03.061.

[153] Zerroukat M, Djidjeli K, Charafi A. Explicit and implicit meshless methods for linear advection–diffusion-type partial differential equations. *International Journal for Numerical Methods in Engineering* 48(1):(2000), 19–35. doi: 10.1002/(SICI)1097-0207(20000510)48:1<19::AID-NME862>3.0.CO;2-3.

[154] Mollerup AL, Mikkelsen PS, Thornberg D, Sin G. Controlling sewer systems – a critical review based on systems in three EU cities. *Urban Water Journal* 14(4):(Mar. 2016), 435–442. doi: 10.1080/1573062x.2016.1148183.

[155] Sahagun L. High Cost of Fighting Urban Runoff Examined in Report. *LA Times*:(2013).

[156] Wines M. Behind Toledo's water crisis, a long-troubled Lake Erie. *New York Times* 4:(2014).

[157] *Dams and Development: A New Framework for Decision-making - The Report of the World Commission on Dams*. Routledge, Nov. 2000. doi: 10.4324/9781315541518.

[158] Arora M et al. Interactions between centralized and decentralized water systems in urban context: A review. *Wiley Interdisciplinary Reviews: Water* 2(6):(June 2015), 623–634. doi: 10.1002/wat2.1099.

[159] Coffman LS, Goo R, Frederick R. Low-Impact Development: An Innovative Alternative Approach to Stormwater Management. *WRPMD 1999*. American Society of Civil Engineers, June 1999. doi: 10.1061/40430(1999)118.

[160] Strecker E et al. Determining urban stormwater BMP effectiveness. *Proceedings of the Water Environment Federation* 2000(6):(Jan. 2000), 2395–2412. doi: 10.2175/193864700785150457.

[161] Askarizadeh A et al. From Rain Tanks to Catchments: Use of Low-Impact Development To Address Hydrologic Symptoms of the Urban Stream Syndrome. *Environmental Science & Technology* 49(19):(Sept. 2015), 11264–11280. doi: 10.1021/acs.est.5b01635.

[162] Mullapudi A, Bartos M, Wong B, Kerkez B. Shaping Streamflow Using a Real-Time Stormwater Control Network. *Sensors* 18(7):(July 2018), 2259. doi: 10.3390/s18072259.

[163] Petrucci G, Rioust E, Deroubaix JF, Tassin B. Do stormwater source control policies deliver the right hydrologic outcomes? *Journal of Hydrology* 485:(Apr. 2013), 188–200. doi: 10.1016/j.jhydrol.2012.06.018.

[164] Kirkby MJ. Tests of the random network model, and its application to basin hydrology. *Earth Surface Processes* 1(3):(July 1976), 197–212. doi: 10.1002/esp.3290010302.

[165] Gupta VK, Waymire E, Rodríguez-Iturbe I. On Scales, Gravity and Network Structure in Basin Runoff. *Scale Problems in Hydrology*. Springer Netherlands, 1986, pp. 159–184. doi: 10.1007/978-94-009-4678-1_8.

[166] Gupta VK, Mesa OJ. Runoff generation and hydrologic response via channel network geomorphology — Recent progress and open problems. *Journal of Hydrology* 102(1-4):(Sept. 1988), 3–28. doi: 10.1016/0022-1694(88)90089-3.

[167] Mesa OJ, Mifflin ER. On the Relative Role of Hillslope and Network Geometry in Hydrologic Response. *Scale Problems in Hydrology*. Springer Netherlands, 1986, pp. 1–17. doi: 10.1007/978-94-009-4678-1_1.

[168] Marani A, Rigon R, Rinaldo A. A Note on Fractal Channel Networks. *Water Resources Research* 27(12):(Dec. 1991), 3041–3049. doi: 10.1029/91wr02077.

[169] Troutman BM, Karlinger MR. Unit hydrograph approximations assuming linear flow through topologically random channel networks. *Water Resources Research* 21(5):(May 1985), 743–754. doi: 10.1029/wr021i005p00743.

[170] Mantilla R, Gupta VK, Troutman BM. Scaling of peak flows with constant flow velocity in random self-similar networks. *Nonlinear Processes in Geophysics* 18(4):(July 2011), 489–502. doi: 10.5194/npg-18-489-2011.

[171] Tejedor A, Longjas A, Zaliapin I, Foufoula-Georgiou E. Delta channel networks: 1. A graph-theoretic approach for studying connectivity and steady state transport on deltaic surfaces. *Water Resources Research* 51(6):(June 2015), 3998–4018. doi: 10.1002/2014wr016577.

[172] Tejedor A, Longjas A, Zaliapin I, Foufoula-Georgiou E. Delta channel networks: 2. Metrics of topologic and dynamic complexity for delta comparison, physical inference, and vulnerability assessment. *Water Resources Research* 51(6):(June 2015), 4019–4045. doi: 10.1002/2014wr016604.

[173] Schubert JE, Burns MJ, Fletcher TD, Sanders BF. A framework for the case-specific assessment of Green Infrastructure in mitigating urban flood hazards. *Advances in Water Resources* 108:(2017), 55–68. issn: 0309-1708. doi: 10.1016/j.advwatres.2017.07.009.

[174] Zhang B, Xie Gd, Li N, Wang S. Effect of urban green space changes on the role of rainwater runoff reduction in Beijing, China. *Landscape and Urban Planning* 140:(2015), 8–16. issn: 0169-2046. doi: 10.1016/j.landurbplan.2015.03.014.

[175] Norton BA et al. Planning for cooler cities: A framework to prioritise green infrastructure to mitigate high temperatures in urban landscapes. *Landscape and Urban Planning* 134:(2015), 127–138. issn: 0169-2046. doi: 10.1016/j.landurbplan.2014.10.018.

[176] Meerow S, Newell JP. Spatial planning for multifunctional green infrastructure: Growing resilience in Detroit. *Landscape and Urban Planning* 159:(2017), 62–75. issn: 0169-2046. doi: 10.1016/j.landurbplan.2016.10.005.

[177] Cattafi M et al. Optimal placement of valves in a water distribution network with CLP(FD). *Theory and Practice of Logic Programming* 11(4-5):(2011), 731–747. doi: 10.1017/S1471068411000275.

[178] Creaco E, Franchini M, Alvisi S. Optimal Placement of Isolation Valves in Water Distribution Systems Based on Valve Cost and Weighted Average Demand Shortfall. *Water Resources Management* 24(15):(Dec. 2010), 4317–4338. issn: 1573-1650. doi: 10.1007/s11269-010-9661-5.

[179] Perelman L, Ostfeld A. Application of Graph Theory to Sensor Placement in Water Distribution Systems. *World Environmental and Water Resources Congress 2013*. American Society of Civil Engineers, May 2013. doi: 10.1061/9780784412947.060.

[180] Yazdani A, Jeffrey P. Robustness and Vulnerability Analysis of Water Distribution Networks Using Graph Theoretic and Complex Network Principles. *Water Distribution Systems Analysis 2010*. American Society of Civil Engineers, Dec. 2011. doi: 10.1061/41203(425)85.

[181] Tzatchkov VG, Alcocer-Yamanaka VH, Ortíz VB. Graph Theory Based Algorithms for Water Distribution Network Sectorization Projects. *Water Distribution Systems Analysis Symposium 2006*. American Society of Civil Engineers, Mar. 2008. doi: 10.1061/40941(247)172.

[182] Hajebi S et al. Water distribution network sectorisation using graph theory and many-objective optimisation. *Journal of Hydroinformatics.*(Aug. 2015). doi: 10.2166/hydro.2015.144.

[183] Ayalew TB, Krajewski WF, Mantilla R. Exploring the Effect of Reservoir Storage on Peak Discharge Frequency. *Journal of Hydrologic Engineering* 18(12):(Dec. 2013), 1697–1708. doi: 10.1061/(asce)he.1943-5584.0000721.

[184] Liu YY, Barabási AL. Control principles of complex systems. *Reviews of Modern Physics* 88(3):(Sept. 2016). doi: 10.1103/revmodphys.88.035006.

[185] Liu YY, Slotine JJ, Barabási AL. Controllability of complex networks. *Nature* 473(7346):(May 2011), 167–173. doi: 10.1038/nature10011.

[186] Ruths J, Ruths D. Control Profiles of Complex Networks. *Science* 343(6177):(Mar. 2014), 1373–1376. doi: 10.1126/science.1242063.

[187] Summers TH, Lygeros J. Optimal Sensor and Actuator Placement in Complex Dynamical Networks. *IFAC Proceedings Volumes* 47(3):(2014), 3784–3789. doi: 10.3182/20140824-6-za-1003.00226.

[188] Yan G et al. Controlling Complex Networks: How Much Energy Is Needed? *Physical Review Letters* 108(21):(May 2012). doi: 10.1103/physrevlett.108.218703.

[189] Yan G et al. Spectrum of controlling and observing complex networks. *Nature Physics* 11(9):(Aug. 2015), 779–786. doi: 10.1038/nphys3422.

[190] Shirin A, Klickstein IS, Sorrentino F. Optimal control of complex networks: Balancing accuracy and energy of the control action. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27(4):(Apr. 2017), 041103. doi: 10.1063/1.4979647.

[191] Czuba JA, Foufoula-Georgiou E. Dynamic connectivity in a fluvial network for identifying hotspots of geomorphic change. *Water Resources Research* 51(3):(Mar. 2015), 1401–1421. doi: 10.1002/2014wr016139.

[192] OCallaghan JF, Mark DM. The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing* 27(2):(Aug. 1984), 247. doi: 10.1016/s0734-189x(84)80047-x.

[193] Tarboton DG. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research* 33(2):(Feb. 1997), 309–319. doi: 10.1029/96wr03137.

[194] Shreve RL. Stream Lengths and Basin Areas in Topologically Random Channel Networks. *The Journal of Geology* 77(4):(July 1969), 397–414. doi: 10.1086/628366.

[195] Rodriguez-Iturbe I, Rinaldo A. *Fractal river basins: chance and self-organization*. Cambridge University Press, 2001.

[196] Tak LD, Bras RL. Incorporating hillslope effects into the geomorphologic instantaneous unit hydrograph. *Water Resources Research* 26(10):(1990), 2393–2400.

[197] Moore ID, Grayson R, Ladson A. Digital terrain modelling: a review of hydrological, geomorphological, and biological applications. *Hydrological processes* 5(1):(1991), 3–30.

[198] Bartos M. Controller Placement Code. https://github.com/kLabUM/hydraulic-controller-placement. 2018.

[199] Bartos M. pysheds: simple and fast watershed delineation in python. https://github.com/mdbartos/pysheds. 2018.

[200] Lehner B, Verdin K, Jarvis A. New Global Hydrography Derived From Spaceborne Elevation Data. *Eos, Transactions American Geophysical Union* 89(10):(2008), 93. doi: 10.1029/2008eo100001.

[201] United States Geological Survey. National Hydrography Geodatabase. https://viewer.nationalmap.gov/viewer/nhd.html?p=nhd. 2013.

[202] Homer C et al. Completion of the 2011 National Land Cover Database for the conterminous United States–representing a decade of land cover change information. *Photogrammetric Engineering & Remote Sensing* 81(5):(2015), 345–354.

[203] Moody JA, Troutman BM. Characterization of the spatial variability of channel morphology. *Earth Surface Processes and Landforms* 27(12):(2002), 1251–1266. doi: 10.1002/esp.403.

[204] *iSWM design manual for site development, Appendix A: Rainfall tables for North Central Texas*. Tech. rep. North Central Texas Council of Governments, Environment and Development Department, 2006.

[205] United States Environmental Protection Agency. ORD Stormwater Management Model. https://github.com/USEPA/Stormwater-Management-Model. 2018.

[206] Muschalla D et al. Ecohydraulic-driven real-time control of stormwater basins. *Journal of Hydrology* 511:(Apr. 2014), 82–91. doi: 10.1016/j.jhydrol.2014.01.002.

[207] Poresky A, Boyle R, Cadwalader O. Piloting Real Time Control Retrofits of Stormwater Facilities: Two Oregon Case Studies and Beyond. *Proceedings of the Pacific Northwest Clean Water Association, Boise, ID, USA*:(2015), 26–27.

[208] Goldman DM. Quantifying Uncertainty in Estimates of Regulated Flood Frequency Curves. *Bridging the Gap*. American Society of Civil Engineers, May 2001. doi: 10.1061/40569(2001)273.

[209] Bartos M, Kerkez B. Hydrograph peak-shaving using a graph-theoretic algorithm for placement of hydraulic control structures. *Advances in Water Resources* 127:(May 2019), 167–179. doi: 10.1016/j.advwatres.2019.03.016.

[210] Kalyanapu AJ, Burian SJ, McPherson TN. Effect of land use-based surface roughness on hydrologic model output. *Journal of Spatial Hydrology* 9(2):(2010).