Real-time Measurement and Control of Urban Stormwater Systems

by

Brandon Preclaro Wong

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2017

Doctoral Committee
Professor Branko Kerkez, Chair
Professor Laura K. Balzano
Professor Jerome P. Lynch
Professor Jeffrey T. Scruggs

Brandon Preclaro Wong

bpwong@umich.edu

ORCID iD: 0000-0002-0109-6150

Dedication

This dissertation is dedicated to my Mom and Dad for their love and support.

Acknowledgements

This work was financially supported by the University of Michigan and the Great Lakes Protection Fund under Project Number 1035.

First and foremost, I would like to acknowledge my advisor, Professor Branko Kerkez, in particular for his mentorship, guidance, and encouragement. Since meeting at UC Berkeley, Branko has been an inspiration and motivational force to always aim higher as an academic and research professional. He has played a key role in my memorable experiences at Michigan and I consider myself lucky to have such a great role model, teacher, and friend.

I would also like to thank my dissertation committee, Professor Jeff Scruggs (CEE), Professor Jerry Lynch (CEE), and Professor Laura Balzano (ECE), for their support and guidance throughout my academic career. Professor Scruggs helped me gain a deep appreciation for linear and stochastic systems and, in general, has been a wonderful influence on my time at Michigan. His input on control systems also helped guide the research on systems-level control in Chapters 5 and 6. I am particularly grateful to Professor Lynch for always somehow finding a spare moment to catch up and to Professor Balzano for helping me believe that I could be an Electrical Engineer.

To the Real-Time Water Systems crew, it wouldn't be the same without you. Kevin Fries has been a great labmate, travel partner, and amazing friend. The friendship and support of Matt Bartos, Abhi Mullapudi, and Sara Troutman, and honorary members including, Will Greenwood, Alyssa Kody, and Paul Beata, have helped these past years fly by.

From deploying the first sensor nodes, to braving the cold and wet weather, to staying out 'til dusk fixing valves, the grand achievements in this dissertation would not have been possible without the dedication, support, and great times from all the students who worked in our lab. Ho-Zhen Chen and Yanglin Zhao were instrumental in the very first deployments. To Rachel Menge, Steve Roudebush, Reece LaFortune, Katya Rahkmatulina, Nick Jeng, Adarash Mishra, Anna Billings, Betty Wan, Jean Yves Ishimwe, Thomas Bejin, Tiernan Slawecki, Gerardo Longora, Isaac Balinksi, Carly Sharp, and Jackie Fortin, I cannot thank you enough.

My time on the Great Lakes Protection Fund project was an experience I will never forget, and Professor Cyndee Gruden and Matthew Lewis always kept me looking forward to the next time we would meet. I would also like to thank Luis Montestruque, Ruben Kertesz from EmNet, LLC and Marcus Quigley from OptiRTC, as well as the EarthCube CHORDS team, Sam Hatchett, Jon Pollack, and Bradley Eck for the engaging conversations around real-time data and the future of stormwater.

I am forever gracious to our collaborators for facilitating and supporting the deployment of our sensor nodes and controllers throughout Ann Arbor, including Ric Lawson, Lauren Burns, and Stevi Kosloskey at Huron River Watershed Council; the Washtenaw County Water Resources Commissioner, Evan Pratt, and Deputy of Environmental Services, Harry Sheehan; Troy Baughman and Jennifer Lawson from the City of Ann Arbor; and John Haling, who was pivotal to the valves installed in Ann Arbor. Our conversations and the lessons learned from the field give tangible purpose to the work in this dissertation.

Finally, to my family for their endless love and support. Tony and Jennifer, thank you rooting me on every step of the way. To my parents, Dean and Carolina, your love and encouragement has lifted me achieve more than I could have ever imagined.

Table of Contents

List of Figures

List of Tables

Abstract

Urban watersheds are being stressed beyond their capacity as storms are becoming more frequent and intense. Flash flooding is the leading cause of natural disaster deaths in the United States. Simultaneously, population pressures are changing landscapes and impairing water quality by altering the composition of urban stormwater runoff. Presently, the only solution to combat these challenges relies on the construction of larger infrastructure, which is cost prohibitive for most cities and communities.

Advances in technology and autonomous systems promise to usher in a new generation of “smart” stormwater systems, which will use city-scale sensing and control to instantly “redesign” themselves in response to changing inputs. By dynamically controlling pumps, valves and gates throughout the entire city this paradigm promises to push the performance of existing assets without requiring the construction of new infrastructure. This will allow for entire urban watersheds to be dynamically controlled to meet a variety of desired outcomes.

Despite technological advances and an established fundamental knowledge of water systems, it is presently entirely unclear how “smart” stormwater systems can actually be built. This dissertation conducts a review of existing “static” solutions and provides an assessment of a number of limited, but highly promising, real-world control studies. An analysis of sensor network scalability is then carried out, focusing on how large water sensor networks can be enabled by leveraging wireless connectivity and web-services. A study of urban water quality follows, which shows how real-time data improve our watershed-scale understanding of pollutant loads during storm events. In turn, through an unprecedented real-world study, it is illustrated how this improved understanding can be used to control flows across a watershed. A feedback control-based approach is then introduced to enable the control of urban watersheds. Through extensive simulation, this framework is applied to identify which control assets have the highest potential to improve watershed performance and to determine how many sites must be retrofitted to achieve desired outcomes. Finally, an analysis of input uncertainty is carried out, which quantifies the importance of weather forecasts in improving control performance across the scale of urban headwater catchments. The dissertation closes by laying out future directions in the emerging field of “smart” stormwater research.

Chapter 1 Introduction

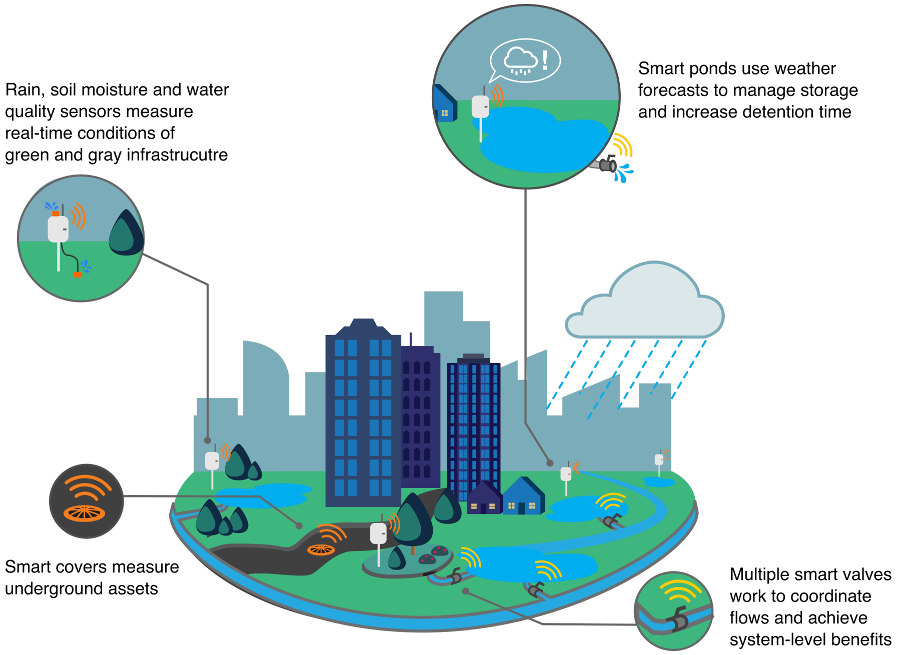

Rain, soil moisture and water quality sensors measure real-time conditions of green and gray infrastrucutre
Smart ponds use weather forecasts to manage storage and increase detention time
Smart covers measure underground assets
Multiple smart valves work to coordinate flows and achieve system-level benefits

Figure 1.1. System-level stormwater measurement and control.

1.2 “Smarter” stormwater systems

Aging urban stormwater systems are being stressed beyond their capacity as storms are becoming more frequent and intense [1-3]. Flash flooding is the leading cause of natural disaster deaths in the United States [4]. Simultaneously, population pressures are changing landscapes and thus altering the composition of urban stormwater runoff. Runoff pollution is readily acknowledged as one of our most pressing environmental challenges [5]. Presently, the only solution to combat flooding and water quality impairments involves the construction of larger infrastructure, which is cost prohibitive for most cities and communities. Traditional infrastructure solutions are inherently static and are unable to adapt to highly dynamic storms and changing landscapes. As such, new solutions are direly needed to improve the stability and health of urban watersheds.

Advances in technology and autonomous systems are promising user in a new generation of "smart" stormwater systems, which will use city-scale sensing and control to instantly "redesign" themselves in response to changing inputs (Figure 1.1). By dynamically controlling pumps, valves and gates throughout the entire city, this paradigm promises to push the performance of existing assets without requiring the construction of new infrastructure. This will allow for entire urban watersheds to be dynamically controlled to meet a variety of desired outcomes. This brings to bear the classic representation of a feedback loop (Figure 1.2), wherein the physical state of the watershed is measured by sensors, which then inform the control logic that ultimately changes the configuration of infrastructure to *push* the system toward a desired state.

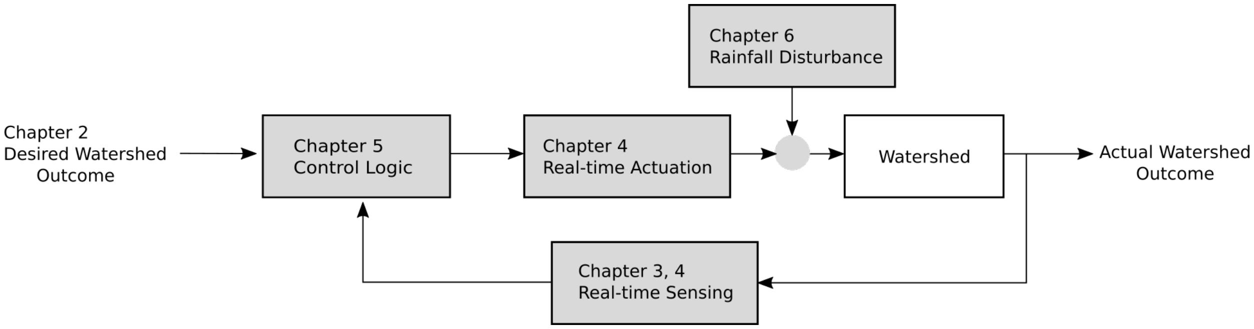


Figure 1.2. The chapters in this dissertation address the major components necessary for the intelligent control of a watershed.

Despite technological advances and our fundamental knowledge of water systems, presently, it is entirely unclear how “smart” stormwater systems can actually be built. System-level control of urban watersheds requires the fusion of various domain expertise, spanning hydrology, water quality, sensing, signal processing, and control theory. As such, a number of fundamental knowledge gaps impede our ability to translate the vision of “smart” water systems into reality:

- We do not understand what benefits, if any, real-time stormwater systems provide when compared to *static* infrastructure solutions.
- We do not understand how recent technological innovations in sensing, communications and cloud computing can be combined to enable massive-scale water sensor and control networks.
- We do not understand the dynamic composition of urban runoff, which impedes our ability to decide how the flow of water should be controlled to improve water quality.
- We do not understand how domain knowledge from hydrology and hydraulics should be placed into a control-theoretic context to allow for system-level control of urban watersheds.
- Finally, we do not understand the extent to which uncertainty, inherent in highly dynamic weather, will affect the performance of watershed-scale control approaches.

Bridging these knowledge gaps will require a systems-level research approach.

1.3 Thesis contributions

The goal of this dissertation is to enable the real-time study and control of urban watersheds. To that end, the specific contributions of this thesis tackle a variety of theoretical and technological challenges that will ultimately underpin a complete framework for the study of “smart” stormwater systems. In summary:

- **Chapter 2**: The contribution of this chapter is a review of existing “static” solutions and an assessment of limited, but promising, real-world case studies that highlight the future promise and research challenges of real-time control.
- **Chapter 3**: The contribution of this chapter is an analysis of sensor network scalability, focusing on how large water sensor networks can be enabled by leveraging wireless connectivity and web-services.
- **Chapter 4**: The contribution of this chapter is a study of urban water quality, which shows how real-time data improve the watershed-scale understanding of pollutant loads during storm events. In turn, we then illustrate, through an unprecedented real-world study, how this improved understanding can then be used to control flows across a watershed.
- **Chapter 5**: The contribution of this chapter is a dynamical feedback approach for the control of urban watersheds. This formulation is used in extensive simulation to identify which control assets have the highest potential to improve watershed performance and how many sites must be retrofitted to achieve desired hydraulic outcomes.
- **Chapter 6**: The contribution of this chapter is an analysis of input uncertainty, which quantifies the importance of weather forecasts in improving control performance across the scale of urban headwater catchments.

In Chapter 2, we summarize the challenges of stormwater infrastructure solutions. Case studies and possibilities for real-time stormwater control are explored, while research challenges and knowledge gaps are identified. The chapter explores how existing stormwater systems require significant investments to meet challenges imposed by climate change, rapid urbanization, and evolving regulations. We then illustrate how there is now an unprecedented opportunity to improve urban water quality by equipping stormwater systems with low- cost sensors and controllers. Most importantly, we outline the most urgent fundamental research challenges that must be addressed before these systems become ubiquitous.

In Chapter 3, motivated by the lack of real-time data and insufficient environmental sensing platforms, we investigate how large sensor and control systems can be realized. While real-time sensor feeds have the potential to transform both environmental science and decision-making, such data are rarely part of real-time workflows, analyses and modeling tool chains. Despite benefits, ranging from the detection of malfunctioning sensors to adaptive sampling, the limited number of existing real-time platforms across environmental domains pose a barrier to the adoption of real-time data. We present an architecture built upon 1) the increasing ability to expose environmental sensors as web services, and 2) the merging of these services under recent innovations on the Internet of Things (IoT). By leveraging recent developments in the IoT arena, the environmental sciences stand to make significant gains in the use of real-time data.

In Chapter 4, we apply the discoveries of Chapter 3 to the study of a real-world watershed in the Midwestern US. Before controlling flows, we focus on controlling sensing resources to improve our understanding of runoff dynamics, the knowledge of which will subsequently be used to inform how control algorithms should be parameterized. An approach to adaptively measure runoff water quality is introduced, focusing specifically on characterizing the timing and magnitude of urban pollutographs. Rather than relying on a static schedule or flow-weighted sampling, which can miss important water quality dynamics if parameterized inadequately, novel Internet-enabled sensor nodes are used to autonomously adapt their measurement frequency to real-time weather forecasts and hydrologic conditions. This dynamic approach has the potential to significantly improve the use of constrained experimental resources, such as automated grab samplers, which continue to provide a strong alternative to sampling water quality dynamics when in situ sensors are not available. Compared to conventional flow-weighted or time-weighted sampling schemes, which rely on preset thresholds, a major benefit of the approach is the ability to dynamically adapt to features of an underlying hydrologic signal. A 28 km^2 urban watershed was studied to characterize concentrations of total suspended solids (TSS) and total phosphorus. Water quality samples were autonomously triggered in response to features in the underlying hydrograph and real-time weather forecasts. The study watershed did not exhibit a strong first flush, and intra-event concentration variability was driven by flow acceleration, wherein the largest loadings of TSS and total phosphorus corresponded with the steepest rising limbs of the storm hydrograph. The scalability of the proposed method is discussed in the context of larger sensor network deployments, as well the potential for improving the control of urban water quality. We conclude with an unprecedented real-world case study, in which this same watershed is then controlled using valves, with a specific objective of reducing stream erosion and sediment loads.

Having verified the real-world promise of real-time control, Chapter 5 focuses on investigating the real-time control of entire urban watersheds. Specifically, through exhaustive simulation we seek to answer the question: Where should urban catchments be retrofitted for real-time control and what performance gains can be achieved compared to passive alternatives? Using model of a complex stormwater network, a linearized dynamical representation is developed and paired with a linear quadratic regulator controller. The chapter identifies which combination of controllable sites best achieve the outcome of minimizing flooding while improving water quality, as informed by the studies in prior chapters. We show that control of every storage asset may not be needed, but rather than a small subset of the overall watershed can be controlled to achieve desired outcomes.

In Chapter 6, we examine the importance of weather uncertainty in the control of stormwater systems. Motivated by limitation of feedback control, we quantify the benefits and challenges of using weather forecasts to prepare watersheds in anticipation of rain events. Using both design storms and real weather data, this chapter tests the ability for system-wide control to reduce flooding, limit flowrates, and maximize the retention of stormwater runoff. We show that forecast integration helps to proactively release captured stormwater to prevent flooding while simultaneously achieving the objectives for flow and retention.

Finally, chapter 7 presents a summary of results, highlights the key takeaways, and poses a number of future research questions to promote the continuation of this work.

Chapter 2 Smarter stormwater systems

Figure 2.1. System-level stormwater measurement and control.

2.1 Introduction

The design of stormwater and sewer systems is based on historical observations of precipitation and land use. These systems are inherently static, requiring significant investments to meet challenges imposed by climate change, rapid urbanization, and evolving regulations. As a result, runoff from urban environments is threatening environmental health by lowering the quality of receiving waters, including fisheries, recreational sites and sources of drinking water. There is an unprecedented opportunity, however, to improve urban water quality by equipping existing stormwater systems with low-cost sensors and controllers. This will enable a new generation of *intelligent* green and gray stormwater networks, which will adapt their operation to maximize water quality benefits in response to individual storm events and changing landscapes.

2.2 Static Solutions to a Dynamic Problem

The vast majority of the world's population resides in or near urban centers, underscoring the need to sustainably manage anthropogenic environmental impacts [6, 7]. Urbanization and land development are disruptive to the hydrologic cycle since they result in an altered, more impervious landscape, which promotes increased runoff at the expense of infiltration and evapotranspiration. While most cities maintain a dedicated stormwater infrastructure, ecosystems near many post-industrial cities in the US are adversely impacted by overflows from combined sewers [8-10]. These overflows have increased due to leaks in aging infrastructure and shrinking municipal budgets.

The increase in the volume, velocity and contaminants in stormwater runoff has caused a crisis in receiving water bodies [11-14]. Harmful algal blooms, associated with anthropogenic inputs of nutrients, have resulted in unsafe drinking water, impaired fisheries and damage to recreational waters [15-19]. As such, managing pollutant loadings from urban stormwater has become one of our most pressing environmental challenges [5, 20].

Expansion and upsizing of *gray* infrastructure are perhaps the most common solutions to coping with increased runoff resulting from changing weather and land use [21]. Aggressive climate adaptation via traditional tools may lead to over-designed gray infrastructure, which conveys water too quickly to streams, leading to floodplain encroachment, increases in runoff volumes, and stream erosion. To preserve stream stability and ecological function, advances in stormwater science are calling for traditional peak attenuation designs to be replaced with those that reduce stream erosion during smaller, more frequent storms [22]. As communities seek more resilient and adaptive stormwater solutions, novel and nontraditional alternatives to new construction must be considered.

One such alternative is provided by *green infrastructure (GI)*, which augments impervious urban areas with pervious solutions such as bioswales, green roofs and rain gardens [23-25]. GI is designed to restore some ecosystem functions to pre-urbanization levels by capturing runoff and contaminants before they enter the stormwater system. These solutions have experienced a significant rise in popularity due to their promise to offer a low impact alternative toward buffering flows and improving runoff water quality [26]. Much research remains to be conducted, however, to test the efficacy and scalability of GI as an alternative to gray infrastructure. To that end, more cost-effective sensing solutions are required to assess the in-situ performance and improve the maintenance of GI [27, 28].

While stormwater systems do change (albeit *slowly*), their design performance is often regarded as static due to limited ability to adapt to changing climate and land uses. More importantly, stormwater solutions are engineered on a site-by-site basis, with little consideration given to ensuring that local benefits are actually adding up to achieve a collective outcome [29]. Rather than offering an alternative, a new solution promises to augment, rather than replace, green and gray infrastructure. This approach relies heavily on sensor and information technology to make existing stormwater systems more adaptive by embedding them with real-time *intelligence*.

2.3 Real-time Adaptive Management

The past decade has witnessed significant advances and reduction in the cost of novel sensors, wireless communications and data platforms. In large, much of this development has accompanied the recent boom on the *Internet of Things (IoT),* a technological movement that promises to build the next generation of interconnected and *smart* buildings and cities [30]. The stormwater sector has been slow in its adoption of these technologies, especially in the context of high-resolution and real-time decision-making. Present uses of sensors range from regulatory compliance [31, 32] to performance studies of individual stormwater facilities [33]. These technological advances have the potential to become highly transformative, however, by enabling stormwater infrastructure to evolve from static to highly adaptive (Figure 2.1). By coupling the flow of water with the flow of information, modern stormwater infrastructure will adapt itself in real-time to changing storms and land uses, while simultaneously providing a highly cost-effective solution for cities that are otherwise forced to spend billions on stormwater reconstruction [34].

Given advances of modern sensors and data acquisition systems, it is now feasible to monitor green and gray infrastructure projects pre- and postconstruction to provide in situ performance metrics. This is afforded by a significant reduction in the cost of sensors and cloud-hosted real-time data systems. Many commercial and open-source platforms, specifically geared toward demands imposed by storm and sewer applications, are now available and promising to lower the cost of wireless sensor deployments. Water flow, stage, precipitation and soil-moisture can now be measured seamlessly and continuously. The development of robust and affordable in situ water quality sensors for nutrients, metals or bacteria is still evolving.

While new measurements will provide significant insight into the study and management of stormwater systems, it is the ability to directly and proactively control these systems that presents the biggest potential impact to water quality. Low-cost, reliable and secure actuators (e.g., valves, gates, pumps) can now be attached to existing stormwater systems to control the flow of water in pipes, ponds and green infrastructure. Examples include inflatable pillows that can be used to take advantage of underused inline storage [35] or smart outlet structures that control water levels in response to real-time data and weather forecasts (Figure 2.2).

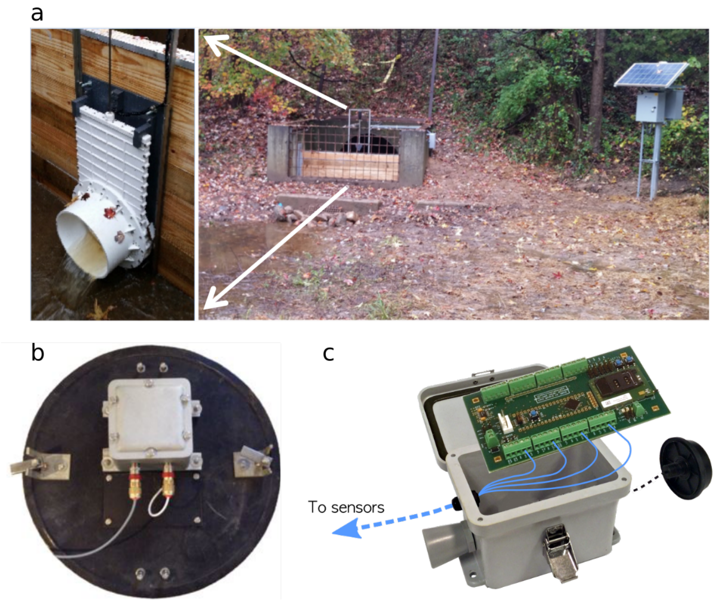


Figure 2.2. Example sensing and control devices. (a) Remote valve for basin control, (b) smart sensing manhole cover, and (c) an open-source sensor node for distributed measurement and control [36].

While real-time process control in water and wastewater treatment has been studied extensively and continues to be a fruitful area of research [37], there is now the opportunity to distribute these treatment ideas to the watershed scale. This presents an exciting new paradigm: *retrofitting existing stormwater infrastructure through cost effective sensors and actuators will transform its operation from static to adaptive, permitting it to be instantly "redesigned" to respond to changing conditions.* There is an inherent complexity associated with control of city-scale systems, however, as they are comprised of a variety of gray and green solutions and driven by complex storm patterns, hydrologic phenomena, and water quality dynamics. The number of studies addressing real-time water quality control is limited but promising, ranging from local- to city-scale control.

2.4 Real-time Control of Individual Stormwater Facilities

Many existing studies focus on the real-time control of stormwater basins and ponds, which are some of the most common elements in a stormwater system [38-40]. Pollutant removal in basins comprises a complex interaction between a number of mechanisms, including sedimentation, flotation, infiltration, biological conversion, and degradation [41]. Traditionally, these facilities are designed as compromises between flood control (detention) and water quality control (retention), with limited ability to adapt functionality to individual storm events. Retrofitting an existing site with a real-time control valve permits it to serve both as a detention and retention basin, as well as a spectrum of in-between configurations. One control rule, for example, opens a valve to drain a pond if a storm is forecasted, which creates additional storage for incoming runoff. Similarly, runoff can be strategically retained after a storm to improve settling and biological uptake. It has been shown, for example, that by temporarily converting a detention basin to a retention basin, the removal efficiency of total suspended solids (TSS) increased from 39% (189 120 g inflow vs 98 269 g outflow) to 90% (e.g., 59 807 g inflow vs 8055 g outflow) and ammonia-nitrogen increased from 10% (101.1 g inflow vs 79.2 g outflow) to nearly 90% (e.g., 163.5 g inflow vs 7.8 g outflow) [38, 41]. Using data from these studied, Mushalla et al. [42] simulated that retaining water using real-time controls may result in up to a 60% improvement in small particle removal compared to a traditional design.

Some studies are also beginning to show that real-time control can play a significant role in removing biological, metal and dissolved contaminants. A controlled basin in Pflugerville, Texas, achieved 6-fold reduction in nitrate plus nitrite-nitrogen compared to the same preretrofit dry basin (0.66 mg/L to 0.11 mg/L) by extending detention time and releasing water before a storm to create additional storage [43]. While biological uptake likely contributed to nitrogen removal, reliable and affordable in situ sensors for many dissolved pollutants are still needed to fully understand the impacts of control to dissolved pollutant removal in natural treatment systems.

Real-time control of a retrofitted detention pond showed that the removal of *Escherichia coli* was improved by strategically retaining water for 24 h after a storm rather than allowing the water to flow though the pond as originally designed [44]. For the controlled basin the outlet concentrations were an order of magnitude lower than inlet concentrations (1940 MPN/100 mL in vs 187 MPN/100 mL out; and 3410 MPN/100 mL in vs 768 MPN/100 mL out), whereas the uncontrolled basin showed limited removal and even increased *E. coli* at the outlet (4350 MPN/100 mL in vs 8860 MPN/100 mL out; 10800 MPN/100 mL in vs 11000 MPN/100 mL out). Since streambed concentrations of *E. coli* were three times higher than in the streamwater, the primary mechanisms for removal were attributed to sedimentation and increased exposure to sunlight. This example also speaks to the need to be cognizant of flow releases from controlled basins, as high outflows can resuspend pollutants. As such, real-time control can be used to modulate the flow rate from storage facilities to reduce downstream erosion and pollutant loads. Such strategies begin to place real-time control into a much broader systems context, whereby each individual stormwater facility not only generates local benefits, but can also be used to improve flow and water quality at the city-scale.

Flow modulation for stream protection was demonstrated at two pilot sites owned by Clean Water Services (CWS) in Washington County, Oregon. In one system (sized to retain 0.2 in. of rainfall), the addition of real-time control to an existing wet pond reduced the volume and duration of channel forming discharges by approximately 25%. In a second facility (a dry detention pond), the use of real-time control was used to minimize release rates in smaller, more frequent storm events while maintaining the ability to match predevelopment peak flows during larger storms. This enhancement was modeled to reduce the volume of erosive flows by nearly 60% and the volume of wet weather discharges by nearly 70% compared to a passive basin (Figure 2.3). Additionally, the use of real-time control increased the average residence time of this facility from 1 to 19 h. In a simulation case study real-time control reduced the required pond volume by 30−50%, compared to a passive facility, while achieving the same level of flow-duration control performance. Finally, based on whole lifecycle cost estimates, it was determined that a real-time control retrofit of an existing stormwater detention facility would be approximately three times lower in lifecycle cost than the equivalent passive alternative [43].

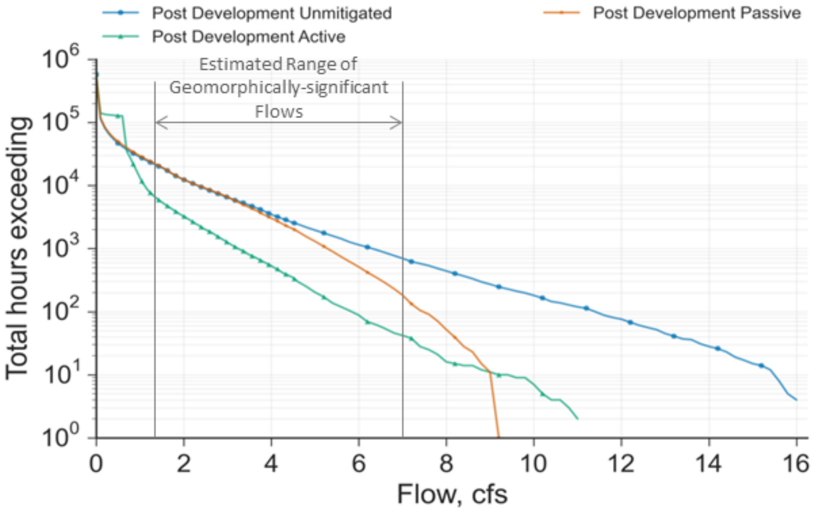


Figure 2.3. Improvement achieved by retrofitting an existing basin to reduce erosive flows.

2.5 Scaling up

An insight into the scalability of real-time control is provided by a large-scale control network that is presently deployed in South Bend, Indiana [45]. The network encompasses 100 km^2 and is comprised of 120 real-time flow and water depth sensors (Figure 2.4), which share information every 5 min. The system has been retrofitted with control valves located at nine CSO regulators to modulate flow into the city's interceptor line. The control valves allow more water to enter the interceptor line when conveyance capacity is available, while avoiding surcharging the interceptor, which may cause surface flooding or structural damage. The system operates by taking advantage of excess conveyance capacity within the interceptor line, which is driven dynamically by spatial or temporal features of specific storms.

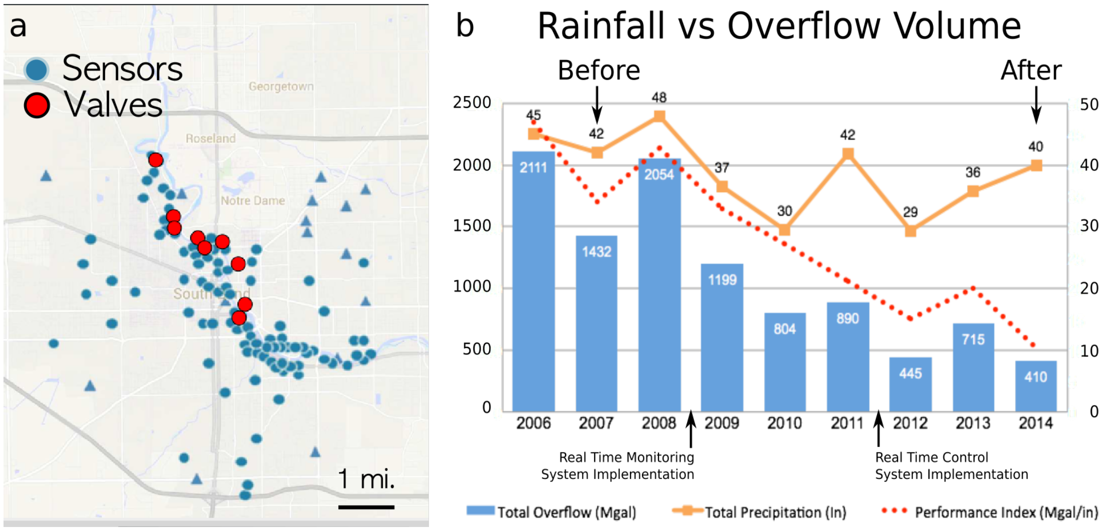


Figure 2.4. Comparison of combined sewer overflows (CSOs) before and after commissioning of real-time sensing and control system in South Bend, IN.

The distributed control strategy uses an agent-based control scheme to optimize the water collection system, whereby each infrastructure component trades its own storage or conveyance capacity to other upstream assets, similar to traders in a stock market [46]. Even before the system was controlled, benefits were achieved by means of monitoring alone. By isolating maintenance issues in its first year of operation (2008), the system helped the utility eliminate critical dry weather sewer overflows, which were occurring an average of 27 times per year. Overall, the control system reduced total sewer overflow volumes from 2100 MGal to 400 MGal from 2006−2014 (Figure 2.4). Even after adjusting for total annual rainfall, a near 5-fold performance improvement (ratio of overflows to precipitation) was achieved. While a reduction in *E. coli* concentrations (443 cfu/100 mL to 234 cfu/100 mL) in the downstream sewer locations was also observed, a more comprehensive ecological study is warranted to study the impacts of real-time control to *E. coli* removal mechanisms. It is estimated that over one billion gallons of untreated sewer flows were blocked from flowing into the river, suggesting that real-time control played a role in improving water quality

2.6 Knowledge Gaps

2.6.1 Systems Thinking

While nascent, research on real-time stormwater control is not limited by technology, but rather by a much more fundamental need to understand the complex spatiotemporal dynamics that govern water flow and quality across large urban areas. One of the largest challenges with existing stormwater solutions relates to their design as single entities. This means that benefits achieved at a local scale may often be masked or eliminated at the city scale if the performance of an individual element is not designed in a broader systems context [29, 47]. Perhaps the biggest benefit of control relates to the ability to leverage real-time interconnection to guide the behavior of individual elements to achieve city-scale benefits.

There is a need to build upon prior and ongoing research efforts on best management practices (BMPs) [24, 33, 48, 49] to understand how individual green and gray stormwater solutions perform when stressed by varying climate, storms, and runoff dynamics. Many studies focus on hydrologic control and removal of solids and bacteria, but much work still remains to be done to determine the impacts of these solutions to the treatment of metals, nutrients and emerging contaminants. This will require the expanded development of cheap and reliable sensors for these pollutants. Furthermore, there is an urgent need to fill a knowledge and measurement gap on the interconnectedness of BMPs across various scales and runoff dynamics (e.g., first flush vs peak flow). By improving the understanding of stormwater networks as a function of scale, it will then be possible to posit how very large systems (ten to a hundred ponds, for example) should be controlled or tuned in real-time to achieve a collective outcome.

2.6.2 Uncertainty

The role of uncertainty is rarely acknowledged in the design of traditional stormwater systems, since it is assumed that many transient system behaviors will average out into a cumulative performance over time. The benefits of real-time control, however, are highly underpinned by uncertainties related to weather forecasts, models, control algorithms, and sensor measurements. Some elements of the system will always remain unmeasured or not understood. Furthermore, many control decisions will continue to be based on hydraulic parameters, such as flow or residence time. Until reliable and low-cost water quality sensors become available, water quality control decisions will rely on statistical correlations or physical models. It will be important to quantify the role of the resulting "error bars" on the performance of real-time control.

As with many controlled systems, there may be an inherent risk to infrastructure, private property, or even human life due to poorly designed control algorithms. Since risk relates directly to uncertainty, reliable and consistent real-time operations can only be achieved by exhaustively quantifying the role of uncertainty in control operations. Furthermore, even the best controllers and sensors may only achieve marginal benefits if storms cannot be predicted adequately, thus calling for the need to begin investigating the value of weather forecasts in control operations. Many other examples can be given, but studies exploring the role of uncertainty have yet to be conducted.

2.7 Outlook and Broader Adoption

Real-time control promises to revolutionize the management of urban water quality by providing the ability to significantly improve the operation of existing stormwater assets. As the community of researchers grows, there will be a need to develop baseline performance metrics, study sites, measurement platforms, and data sets. Research on stormwater capture and direct use (reuse) has recently increased [50] due to the potential of reclaimed stormwater to serve arid regions. In drought-prone regions of the U.S., where stormwater direct use is becoming one of the few viable water recovery options, sensing and real-time control will improve stormwater extraction compared to static or natural treatment options. Controlling the timing and magnitude of flows and improving removal of contaminants before they reach the plant will also result in a reduction in resources required for treatment in combined sewer systems.

Outfitting stormwater infrastructure with sensors and digital control systems introduces new opportunities for efficiency and new risks of failure. Responsible use of these systems extends beyond deployment, requiring ongoing effort to maintain trust in the data produced and the integrity with which control actions are followed. As with all Best Management Practices [24], standards will be required to facilitate broader adoption of real-time control and to assess the risks introduced by the use of information sourced from these embedded systems. Future standards may focus around data formats, sensor requirements or actuator specifications, and will need to ensure interoperability between various sites. Failure to recognize, plan for, and manage the ongoing cyber security risks introduced by the distributed installation of sensors and actuators in stormwater infrastructure will result in new risks to public health and safety, which may undermine trust in broader efforts to deliver the potential benefits of these technologies.

There will be a need to address regulatory compliance, ownership, governance, and operational jurisdictions relating to real-time controlled systems. Unlike existing deployments of sensor and control systems in wastewater treatment, digital stormwater infrastructure is deployed across a watershed, outside of buildings staffed by an operations team. A key tension relates to jurisdiction, both in terms of who owns the infrastructure being controlled and which software system provides this dynamic capacity. Many cities may only wish to try retrofitting some sites, with the plan to augment their systems over time as they see benefits. This raises the possibility that many software systems may operate simultaneously and interfere with a global goal. If control systems are deployed by a spectrum of public and private stakeholders, they should nonetheless interoperate to provide capacity for watershed-scale control and maintenance. Governance models must be explored to facilitate cooperation and liability concerns. While solutions to these concerns can build on successful models used for ownership and operation of passive controls, they may require further thought in their translation to real-time controlled systems.

Beyond technical challenges, the ecosystem of municipalities and engineering firms must adapt to accommodate real-time control within a large umbrella of green and gray infrastructure solutions. Broader community engagement is necessary to facilitate dissemination and adoption of real-time stormwater control. Compliance regulators, such as state and federal environmental protection agencies, must be highlighted as members of this community, since many cities are wary of innovation because of perceptions that regulators will reject nontraditional solutions. Environmental consulting firms, municipalities, and researchers will need to acquire nontraditional skillsets, which span electrical engineering and computer science. To help with this effort, a major initiative is presently underway to organize an open-source consortium and share reference implementations on real-time stormwater control (http://open-storm.org). While open-source options for sensing and control are alluring due to their perceived cost, examples of holistic open-source approaches, which integrate environmental science, technology and engineering design, have yet to be developed. To that end, this consortium will serve as a hub for reference applications, standards, architectures, sensors, hardware and algorithms, to show that it is well within the abilities of most academic groups, municipalities and engineering firms to begin instrumenting and controlling stormwater infrastructure.

Chapter 3 Real-time environmental sensor data: an application to water quality using web services

3.2 Introduction

Before control can be carried out, real-time measurements have to be made first, which poses significant challenges considering the size and complexity of urban watersheds. Recent advances in sensing, computation and communications have enabled a massive suite of low-cost, low-power connected devices. This is particularly true for modern wireless sensor networks [51, 52], which now support the reliable, low-cost, near-instant transmission of measurements from field-deployed sensors. For enterprise-scale web applications, *RESTful* web services have also witnessed a surge in popularity [53] while advances in the hardware realm have been accompanied by new architectures and protocols that exploit the bidirectional communication and Internet-connectivity of embedded devices. As such, libraries and *application programming interfaces* (*APIs*) enable users to quickly deploy RESTful web services on almost any software or hardware platform. This is significant, as most new devices from popular hardware and datalogger manufacturers increasingly support web communication via Wi-Fi, Ethernet, cellular, and other physical channels. Through these efforts, the *Internet of Things* (*IoT*) has recently been proposed as the backbone that will route and manage the vast quantities of data collected by these sensor networks [54, 55]. In many environmental applications, however, these technological advances merely serve as a convenience to reduce field visits, provide data visualization, and simplify data collection by streaming sensor data to central repositories for subsequent analysis. Real-time data are rarely used as part of automated workflows, analyses and modeling tool chains.

In the computer science communities, in particular the area of embedded systems, the definition of real-time carries with it explicit performance guarantees, such as deadlines and timing constraints [56]. Such a strict definition, however, may be too technical to appeal to the broader environmental communities. While an actual definition may be out of reach considering the diversity of applications in the environmental sciences, an underlying principle persists: *real-time data* are data available for use as soon as they are collected to make a decision within a constrained time window, independent of sampling frequencies. This principle does not seek to distinguish between notions of real-time or near real-time, as is often the case in many studies [51, 57, 58].

While not ubiquitous across the broader environmental domains, the use of real-time data for decision-making is not novel in some fields. For example, in the atmospheric sciences, satellite data is assimilated daily into advanced models which are used by various scientists and decision makers [59], while across meteorology, real-time radar feeds and terrestrial sensors inform stake holders across agriculture, transportation and disaster response [60]. However, despite the availability of low-cost, low-power hardware and data platforms, the benefits of these real-time resources have yet to be leveraged broadly across the remaining environmental sciences.

Scientific data analyses are more commonly conducted after an experiment has been completed, which for many studies could last months or years. A reason for the lack of real-time data adoption relates to the fact that most scientists may simply be satisfied with *continuous,* rather than *real-time,* data. The use of sensors across the environmental sciences thus appears to be retroactive, rather than adaptive. This would suggest that the major benefit of real-time data relates to decision-making, where assimilation of sensor information into models will enable rapid response to extreme events such as floods, wildfires and earthquakes.

While the ability to respond to natural disasters is invaluable, significant benefits of real-time data arise to environmental researchers as well, especially in the detection of faulty sensors and data acquisition systems. This is particularly true for experiments in harsh or remote environments where site visits may be infrequent and equipment outages can result in significant lapses in continuous data streams. For such experiments, real-time alerts will go a long way toward improving the quality of continuous data sets.

Perhaps the most compelling benefit of real-time data relates to the ability to usher in a new generation of adaptive scientific experiments. By adding real-time functionality to non-real-time studies, scientists will be able to perform innovative studies that respond to dynamic experimental conditions. As illustrated in this chapter, this includes the ability to guide an experiment in real-time to adaptively sample signals or locations of interest during the most relevant intervals, which will significantly improve the use of constrained experimental resources and thus the quality of scientific experiments.

Across many domains, the notion of real-time is often complicated by operational requirements, which drive a lack of consensus around the definition of the actual term. Regardless of application, however, the utility of real-time data is governed by constrained time windows during which decisions have to be made. These time windows can range from days (e.g. climate modeling and data assimilation [57]) to minutes or seconds (e.g. flood or wildfire forecasting [58]). Outside of these time windows the data can be classified as *historical*, thereby limiting their utility for immediate decision-making. A wealth of tools have been developed to store, process and visualize historical sensor data [61-65], but these frameworks have yet to be extended to provide real-time functionality.

In this chapter, we present a summary of existing efforts to enable the use of real-time data across a broad set of domains, showing that the complexity and limited number of these existing real-time data platforms limits their adoption by the environmental sensing community. The majority of these platforms requires persistent expert support and cannot always be easily ported to existing field equipment and sensor networks, even by experienced researchers who readily operate continuous sensing campaigns. With real-time data systems also come different operational requirements, including the ability to continuously update and operate on new data, communicate with remote sites, monitor the operational status of devices, and manage user privileges throughout the system. We discuss these barriers to adoption and present a solution built upon two cornerstones: 1) the shift of environmental sensors and actuators[1] to a more

[1] Sensors generate an electrical signal in response to stimuli from the environment. Actuators respond to an electrical signal and act upon their environment (e.g. a gate that opens or closes).

generic web service model, and 2) the merging of these services under the recent architectural innovations on the Internet of Things.

To that end, we introduce a web service-centric approach to enable a flexible, reliable and powerful means by which to store, transmit and analyze real-time data. By focusing on recent advances in the IoT arena, we will show that the environmental sciences stand to make rapid gains in the use of real-time data while simultaneously improving flexibility related to implementation and maintenance. Rather than building a new platform, we will show how existing IoT platforms already provide a backbone to integrate real-time data from web-enabled environmental sensors and devices to meet requirements of interoperability, support, reliability, and security. By leveraging the services provided by these platforms, these web-enabled sensors and devices can also seamlessly interact with a multitude of web resources, including powerful cloud computing services and web-based models. A use case from the hydrologic sciences illustrates how a script can be deployed as a web service within this framework to enable low-power sensor networks to adaptively sample dynamic water quality parameters during storm events. While not a one-size-fits-all solution, our approach is expected to conform well to the requirements of most environmental applications, particularly for those where large sensor networks are deployed.

3.2.2 Existing platforms and real-time data efforts

Data systems employed across the environmental domains may be broadly classified into two groups: 1) systems used for the storage, retrieval and visualization of data, and 2) data systems designed explicitly for real-time operations. While the former do not explicitly treat real-time data, they do provide powerful mechanisms by which to standardize data retrieval and storage [51, 61, 63-65]. Some of these platforms conform to a set of community standards (e.g. *WaterML*, *DelftFEWS,* etc., see [66]) that reduce operational overhead and enable the seamless use of standard-compliant tools for scalable storage, management and visualization of data. However, interactions with data are often carried out through direct user queries to the system, with no or limited mechanisms in place to automatically notify users of new readings or events as they occur. Furthermore, such architectures are not typically designed to enable alerts or the discovery and access to field-deployed sensors or actuators, thus limiting their use in control-centric and decision-making applications.

A number of these systems are also designed for domain-specific applications, thus limiting their use across a broader set of domains. In most cases, end-users are required to implement and host these real-time systems, which introduces deployment and maintenance complexities in addition to those inherent to deploying and maintaining field-deployed sensors and actuators. This includes, but is not limited, to setting up dedicated servers, installing necessary tool chains, adopting specific programming languages, and guaranteeing system up-time.

A number of platforms have been designed to explicitly treat real-time data. *DataTurbine* [67] is an open source, *Java*-based platform for managing and transporting data from sensor networks and video feeds. Designed for environmental applications, DataTurbine implements a *ring buffer*, much like a size-limited first in, first out queue, to temporarily store the most recent sensor data and reliably route data streams to visualization and storage modules. Given the emphasis on streaming, high data rates can be supported (over 16 Msamples/sec). DataTurbine was developed as a generic streaming data middleware for real-time data acquisition systems, independent from a specific application niche. Some uses of DataTurbine include oceanic studies, climate change research, earthquake engineering, and lake monitoring [68]. However, users are required to run individual DataTurbine services on all servers and field-deployed devices, which limits the number of supported data loggers and controllers to those compatible with Java. In addition to the complexities of setting up a monitoring system, the explicit emphasis on a particular programming language makes it difficult for users unfamiliar with Java to deploy the platform.

An extension to DataTurbine is *Wavellite*, an open source Java suite that supports real-time situational knowledge of heterogeneous datasets and observations [69]. The software interprets data as it streams in by using a suite of machine learning algorithms. One example study using Wavellite applied artificial neural networks to process aerosol and weather data to identify and characterize the formation of particles that could act as cloud condensation nuclei [69]. While powerful, the system is designed to support specialized operations and exhibits limited storage support. Moreover, since Wavellite is built upon DataTurbine, its deployment requires implementation expertise of both platforms.

IBM’s *InfoSphere Streams* is another real-time data analysis tool chain that enables the rapid analysis of real-time data feeds before data is saved into databases [70]. The *Streams* tool chain has been applied across a broad set of industries, including financial services and transportation, to continuously use machine learning to extract information for decision-making. However, given its emphasis on machine learning and its current price point (thousands to tens of thousands of dollars), this tool chain appears to be primarily geared toward larger groups and companies and may thus be out of reach of smaller scientific research groups.

One of the most established real-time data systems is UNIDATA's *Local Data Manager* (*LDM*) [71]. LDM is a package of UNIX-based modules designed for event-driven applications, particularly those relating to atmospheric science data. Users must however host their own servers and setup any relevant UNIX-based tool chains before installing and maintaining LDM. Additionally, porting the system to domains outside of atmospheric sciences introduces further complexity, which may deter use by a broader community.

Another popular real-time data system is *Antelope*[2], which supports a number of language interfaces, including *C, Fortran. Perl, Python* and *MATLAB*. Originally designed for storing and streaming seismic data, the system is also built upon a ring buffer and accompanied by a suite of signal processing tools to analyze waveforms and detect events. The suite of tools resembles a real-time signal-processing platform targeted towards seismic applications. While some broader communities may be too unfamiliar with signal processing and its nuances to adopt this system, the suite of seismological tools may also be too specialized for those seeking more general real-time data functionalities.

3.2.3 Overcoming barriers to real-time data adoption

To address an increasing interest in real-time data applications across the environmental sciences, a working group was formed under the broader umbrella of the U.S. National Science

[2] http://www.brtt.com

Foundation's *EarthCube*[3] initiative. The EarthCube initiative was launched in 2011 to discover *transformative concepts and approaches to create integrated data management infrastructures across the geosciences*. A real-time data workshop was organized in 2013 to determine the needs experienced by a broad spectrum of scientific groups [72]. Discussions and surveys nearly unanimously reported that while there was significant interest in real-time data, users were unsatisfied with the limited set of existing tools, citing their complexity and ease of use (or lack thereof) as a major barrier to adoption[4].

When deploying environmental sensors, the resources required to program firmware and maintain hardware already pose significant demands on research groups. Substantial additional overhead is incurred if real-time functionality is desired. Existing real-time data platforms impose significant requirements in the form of system architectures, operating systems, programming languages, and even sensing platforms, which makes their deployment labor intensive, even for users who already maintain sensor networks for continuous data. For example, *DataTurbine* requires users to manually start and maintain Java instances of the software both as servers and on field hardware, while packages such as LDM require users to compile binaries from source code to match their specific UNIX-based environments. To that end, physical protocol compliance has been proposed as a means of tying into these systems and to reduce implementation overhead. In one study, the authors suggest that to enable the use of their platform [73] sensors should conform to a standard hardware interface, in particular an Ethernet port. Field experience and the sheer variety of sensor platforms significantly undermine

[3] http://www.nsf.gov/geo/earthcube

[4] Efforts are now underway to further study these findings and investigate real-time architectures under the EarthCube Cloud Hosted Real-time Data Services (CHORDS) project. http://earthcube.org/group/chords

the real-world applicability of such requirements and further illustrate the disconnect between those deploying sensors and those designing data platforms. These and other usability constraints raise the barrier to real-time data adoption by enforcing non-trivial design and implementation challenges on users.

The adoption of real-time data across the environmental sciences hinges upon the resolution of a number of broader challenges:

- *Interoperability*: Existing real-time data platforms impose non-trivial requirements on users. Real-time data systems should be designed to permit users to retain their existing toolchains and hardware platforms inasmuch as possible without imposing major additional requirements to maintain servers, compile libraries, or support specific hardware interfaces.
- *Support for real-world, low-power devices*: Sensor selection should be governed by the application and should not be limited by the capabilities of the underlying data infrastructure. Due to power constraints, real-world, battery-powered sensing platforms must duty cycle their web connectivity in exchange for battery life [51, 52, 58, 74]. In such instances, it is unreasonable to expect persistent bi-directional communications between sensors and data services. To further limit the power draw from network communications, data must be transmitted as size-efficiently as possible. A real-time data framework must develop means by which to interact with such devices, balancing intermittent transmissions and wide-ranging bandwidths.
- *Reliability and usability*: Many presently existing real-time platforms and open source projects lack the infrastructure and reliability to support a large user base, including

novice and technically savvy users alike. While these systems are being improved, we contend that reliable and feature-rich commercial platforms should be considered inasmuch as possible to allow those deploying sensors to leverage enterprise-scale reliability on their projects. This will permit users to focus on applying their domain knowledge towards developing applications and experiments, rather than data system design and administration.

- *Security*: Given the nature of real-time data, proper security measures must be taken to ensure that the streams of data from sensors and control of devices are protected via modern encryption and authentication techniques. Existing real-time platforms, such as DataTurbine and UNIDATA's LDM, recommend limiting web connections to specific trusted IP addresses and encrypting packets using a digital signature. However, keys to read, write, create, and delete web resources are not implemented in these platforms, and neither is *HTTP Secure* (*HTTPS*), a common protocol used for information-sensitive web applications such as email and online banking.

Rather than enforcing highly specific hardware and software requirements, web services are emerging as a powerful and versatile interfacing mechanism to connect disparate sensors, data sources and models [62]. The availability of reliable, low-cost wired and wireless technologies has increased significantly over the past decade, permitting most field devices to be connected to the Internet [75, 76]. New *Internet Protocol* (*IP*) addressing schemes are currently shifting from the traditional *IPv4* addressing to *IPv6* addressing, which will ensure any device or service can be uniquely identified and accessed [77, 78]. This is particularly useful for wireless sensor networks, such as those based on Ethernet, mesh, Wi-Fi or cellular protocols, where such networking platforms are shifting toward IPv6-based connectivity to accommodate larger network deployments [77]. As the scale of sensor networks continues to grow, requiring IP connectivity for real-time data applications is not just realistic but inevitable. While some of the existing or legacy commercial hardware platforms may not directly support Internet connectivity (TCP/IP), they can either be equipped with communications modules or will soon be replaced by modern platforms that support remote data access. Those platforms that ultimately do not adopt Internet connectivity may still perform well for applications requiring continuous, but not real-time data. As such, in the building of real-time data applications, a paradigm shift toward web services becomes a reasonable requirement for users wishing to take advantage of modern data platforms and cloud-hosted services.

3.2.4 Leveraging IoT Platforms

Platforms from the *Internet of Things (IoT)* community have recently arisen in response to the rapidly increasing number of wireless devices, which are becoming ubiquitous in the measurement and automation of residential and industrial processes. IoT platforms have been designed to support sensor discovery, real-time data routing and remote device control. Rather than writing firmware code, users can use popular high-level languages, such as Python or MATLAB, to analyze data and actuate a remote device. Much of this logic can even be set up using configurable services with little to no additional programming required (e.g. trigger an alert if a sensor value exceeds a threshold). Users can directly subscribe to real-time data feeds or notifications, which are used to monitor the state of field devices and provide a means by which to transmit alerts. Despite their significant adoption across the sensor and automation communities [54, 77], IoT platforms have yet to penetrate a user base within the environmental domains. A variety of both open source and commercial IoT platforms exist (Table 3.1). When comparing the platforms of the IoT ecosystem, their core functionalities can be broken down into administrative features, transfer protocols and data management.

	Free Plan	Data Logging	Public/Private Access	Multiple Keys	REST Protocol	WebSockets	CoAP\MQTT Protocol	API Library	JSON	XML	CSV	Limited Data Retention	Metadata	Notifications, Warnings	Mobile Sensors	Visualizations	Widgets
COMMERCIAL																	
Amazon IoT		√	√	√	√		√	√	√				√	√	√	√	
Beebotte	√	√	√	√	√	√	√	√	√			√	√		√	√	√
Exosite	√	√	√		√	√	√	√	√		√		√	√		√	√
GroveStreams	√	√	√	√	√				√			√	√	√	√	√	√
Open.Sen.Se	√	√	√		√				√							√	√
SensorCloud	√	√	√		√											√	
Ubidots	√	√		√	√			√	√			√		√	√	√	√
Xively	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	
OPEN SOURCE																	
KSDuino	√	√	√					√				√				√	
Nimbits	√	√	√	√	√				√					√	√	√	
phant.io	√	√	√					√	√	√	√					√	
ThingSpeak	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	√

Table 3.1. Comparison of IoT Platforms.

Each IoT platform supports data exchange via web services and a set of standard protocols, the majority of which relies on RESTful data transfer. Some platforms also support more modern, low-power protocols alternative to HTTP, such as MQTT and CoAP [79, 80], which provide additional sensor-centric functionalities and are designed to improve device battery lifetimes by optimizing the size of packet headers. While raw or comma-separated data formats can be exchanged between web services and devices, the majority of platforms support APIs and formats built around popular framing protocols such as *JavaScript Object Notation* (*JSON*) or *Extenstible Markup Language* (*XML*) [81, 82], which permit for rapid integration with various programming languages and tools. The content and syntax of the payload may vary based on the IoT platform, but many platforms, such as *ThingSpeak* and *Xively,* support a broad selection of operating systems and languages. Many platforms also provide APIs that encode data into the required payload syntax, thus reducing the amount of software development required of the user. As such, the payload is similarly encoded amongst these platforms and interoperability between IoT platforms can be achieved through relatively straightforward content mappings. These APIs are available even for low-level languages, such as C, which permits them to be ported to low-power microcontrollers and data-loggers. For those deploying sensor hardware, the steps to connect to an IoT platform involve a relatively small addition to already existing code. At its simplest implementation, this involves opening a TCP/IP port and transmitting a relatively intuitive JSON-encoded or *Comma Separated Values* (*CSV*)-encoded string. While the CSV encoding is relatively self-explanatory, the JSON string can often be generated on the IoT platform's website and pasted into the low-level code. For those wishing to support remote actuation of a field-deployed device, a callback function can also be implemented to parse and respond to messages received by the IoT platform.

All of the major IoT platforms support administrative control via public and private key access. Field-deployed devices and other web services (e.g. models, visualization services, etc.) that interact with these IoT platforms must be authenticated via these keys to access the services provided by an IoT platform. Key permissions can be generated and configured on the platform without needing to change device firmware. To our knowledge, such authentication systems are rarely implemented on existing real-time environmental data systems (LDM, Antelope, etc.), where access is instead controlled by less complex measures such as limiting connections to specific IP addresses or digitally signing packets with a shared password. Presently, however, few IoT platforms support multiple keys and permissions, which is particularly useful where one would like to limit control of specific resources for certain users. For example, depending on their role, a group of users may be allowed only to subscribe to and view real-time data from devices and may not be permitted to control sensors and actuators. With tiered authentication systems, one can dictate which web resources are available publicly or privately.

To varying degrees, IoT platforms support the long-term storage of data and metadata, thus serving both as real-time data and historical data platforms. Some platforms only focus on limited data handling and storage, primarily serving to buffer data for decision-making, actuation, and alerts. While storage may not be an explicit requirement for most real-time applications, long-term storage can be achieved by routing real-time feeds from an IoT platform into community-maintained and domain-specific databases and storage facilities (e.g. The

CUAHSI Water Data Center[5]). Commercial IoT platforms operate on service-based models [83], where providers launch, scale and maintain the platform, allowing users to focus on the actual applications. Most platforms support a free plan, which permits small projects (1-50 sensors) to leverage the services without a fee. Typically, such plans limit the number of sensors and the frequency at which data can be transmitted. In the case where applications require additional sensors or data transmission at higher frequencies, commercial platforms offer very affordable pricing plans, where fees are charged on a per-use basis. For example, at the time of this study, Amazon's IoT Service[6] supported the transmission of one million messages (512 bytes per message) at a cost of $5 USD, which falls well below the cost that a small scientific group would have to expend on developing, hosting and maintaining a comparable data service.

While the open source platforms are powerful, the benefits of the commercial, enterprise-grade data services cannot be discounted. Most open source platforms must be installed and maintained by the user whereas commercial platforms are oftentimes cloud hosted and can be readily accessed within a few mouse clicks. As with any open source platform, significant expertise and resources are required to ensure robust functionalities that include, but are not limited to, routing feeds, issuing alerts, storing data, adhering to protocols and standards, and coordinating security and user privileges. Although open source versions of these platforms offer such features, it may be unrealistic to assume that all scientific users or decision makers have the expertise or resources to deploy and maintain these complex systems.

[5] https://www.cuahsi.org/wdc

[6] https://aws.amazon.com/iot

We contend that the burden of hosting and maintaining complex, real-time web service architectures should not be offloaded to the user, but, where feasible, should be deferred to reliable hosting providers. Most recently, this has been the case with the paradigm shift toward cloud computing, where commercial computing services are replacing local hosting and computations in various applications [84], including many across the environmental domains. A similar paradigm shift in real-time environmental data services is needed. Commercial sensor data services are hosted and maintained by experts, permitting users to launch an instance at their convenience and focus on their applications rather than system administration. It should be noted that in this chapter our goal is not to advocate any IoT platform in particular, but rather to promote their broader adoption.

3.3 Use case

Motivated by the need to improve our understanding of water quality in streams and rivers, our specific objective was to better understand the dynamics of nutrient loadings to urban and agricultural sources. Large nutrient loads are considered the primary cause of harmful algal blooms and *dead zones* witnessed most recently in the Great Lakes [85] and Gulf of Mexico [86]. Despite the insight provided by dynamic models, our understanding of these nutrient loadings and their origins is still limited significantly by a lack of real-world data. As such, more measurements are needed to resolve the spatiotemporal dynamics of nutrient fluxes into aquatic ecosystems.

3.3.1 Challenges in the measurement of water quality

Compared to water flow, water quality parameters are still relatively difficult to measure [87, 88]. This is particularly true of some water quality constituents (e.g. nutrients, metals and bacteria), where in-situ sensors either do not exist or are too cost-prohibitive to be deployed at meaningful resolutions. In such instances, samples collected by automated samplers are an effective alternative to in-situ measurements [89]. When triggered manually or through a timer mechanism, these samplers actuate a motor to pump a water sample directly from a stream into one of a number of available bottles that are later taken to a laboratory for analysis. Since each sample is time stamped, the laboratory results can then be used to correlate water quality parameters with known physical characteristics or measurements taken by accompanying in-situ sensors.

The number of sample bottles is limited in an automated sampler and its power consumption is very large due to its motorized mechanical components. Given these resource constraints, it becomes necessary to optimize sampling times and frequencies to capture events of interest. Sampling too fast can cause the number of available samples to be depleted before an event is captured. Furthermore, "wasted" samples occur when a storm does not happen as predicted, while "missed" samples occur if the duration of a storm is longer than anticipated. Most often, events of interest include storms, which can cause significant quantities of surface water to flow into nearby streams and rivers, thereby discharging nutrients that have accumulated on land. Nutrient loadings during the beginning of a storm, or the *first flush*, are often considered an indicator of the effect of nearby land use practices on water quality. When measuring the flow of water in a stream or river, the first flush is often evident as a spike in the hydrograph signal [90, 91]. To capture these events, automated samplers have mainly been used on an as-needed basis, where units are placed on a site in anticipation of storm events and programmed to take readings at regular intervals. The feasibility of using automated samplers thus becomes burdensome in terms of cost, battery consumption and manual labor.

3.3.2 Adaptive sampling of hydrologic signals

The drawbacks of automated samplers can be minimized through *adaptive sampling* [92, 93] where, rather than evenly sampling a signal, a controller or algorithm persistently updates a model of a phenomenon using real-time data and then samples only during events of interest. In our approach, the algorithm was a configurable web application that queried a public weather forecast to determine the probability of impending precipitation. The algorithm monitored the hydrograph signal in real-time to determine sudden state changes, such as a rapid rise in the hydrograph. A rule-based optimization procedure was then used to determine when to take the next sample. A theoretical description and evaluation of the algorithm are given in [94]. The algorithm encoded the objective of minimizing the samples required to characterize the first flush of the hydrologic catchment by triggering a sample of water quality to be taken right before a storm (based on weather predictions), a number of samples during the rising limb and inflection points of the hydrograph, and a smaller number of samples following the hydrograph recession. As such, the algorithm guided the automated sampler to respond to both weather forecasts and changes in measured flow values. The implementation assumed a sensor node equipped with an Internet connection, which received sampling commands from a web application.

3.3.3 Hardware

A water quality sensor node was developed using the NeoMote wireless sensing platform [95]. This FPGA-based platform (Cypress PSoC5LP) is programmed in C and features an ultra-low power ARM-Cortex M3 microprocessor, 20-bit low-noise analog to digital converter, configurable on-board storage via an SD card, and variable, low-noise power supplies for sensors. Given the urban study site (Ann Arbor, MI Lat. 42.264855, Lon. -83.688347), cellular coverage was readily available, which thus enabled the use of a low-cost IP-enabled cellular module (Telit CC864-DUAL) for Internet connectivity. The NeoMote platform consumed an average current of 30 micro-amps. The cellular module consumed significantly more, requiring nearly 200 milli-amps during transmission events. To conserve power, the cellular module was duty cycled, where power was cut entirely to the module when it was not being actively used to transmit sensor readings. With the addition of a low quiescent current (1 micro-amp) lithium-ion solar charge controller and a solar panel, the sensor node was designed to operate for years without the need for battery replacements or line power.

The sensor node (Figure 3.1) was interfaced with an automated sampler (ISCO 3700) using a transistor-transistor logic interface (TTL). The automated sampler had a 24-bottle capacity, a standby current of 10 milli-amps and an energy consumption of 2 Amps at 12 VDC during sampling. Even with duty cycling, the use of the automated sampler provided the largest constraint on battery resources, further emphasizing the need to limit sampling to only events of interest. A suite of hydrologic sensors was also attached to the data logger, including an ultrasonic depth sensor (MaxBotix MB7384) and a pressure transducer (Solinst 3001 Levelogger® Edge) for stage measurements. The data logger interfaced with the ultrasonic sensor in TTL serial mode, the pressure transducer via SDI-12, and the conductivity sensor via analog output. The data from the ultrasonic sensor was used to derive an estimate of the hydrograph stage, which was then used by the adaptive sampling algorithm to determine when to trigger the next water quality sample. Data from the pressure transducer was initially used to verify the hydrograph estimates derived by the cheaper ultrasonic sensor as a means of vetting its use for future studies. While not used in this study, an analog conductivity sensor (Campbell Scientific CS547A) was also connected to the platform to assess benefits of triggering water quality samples based on conductivity thresholds.

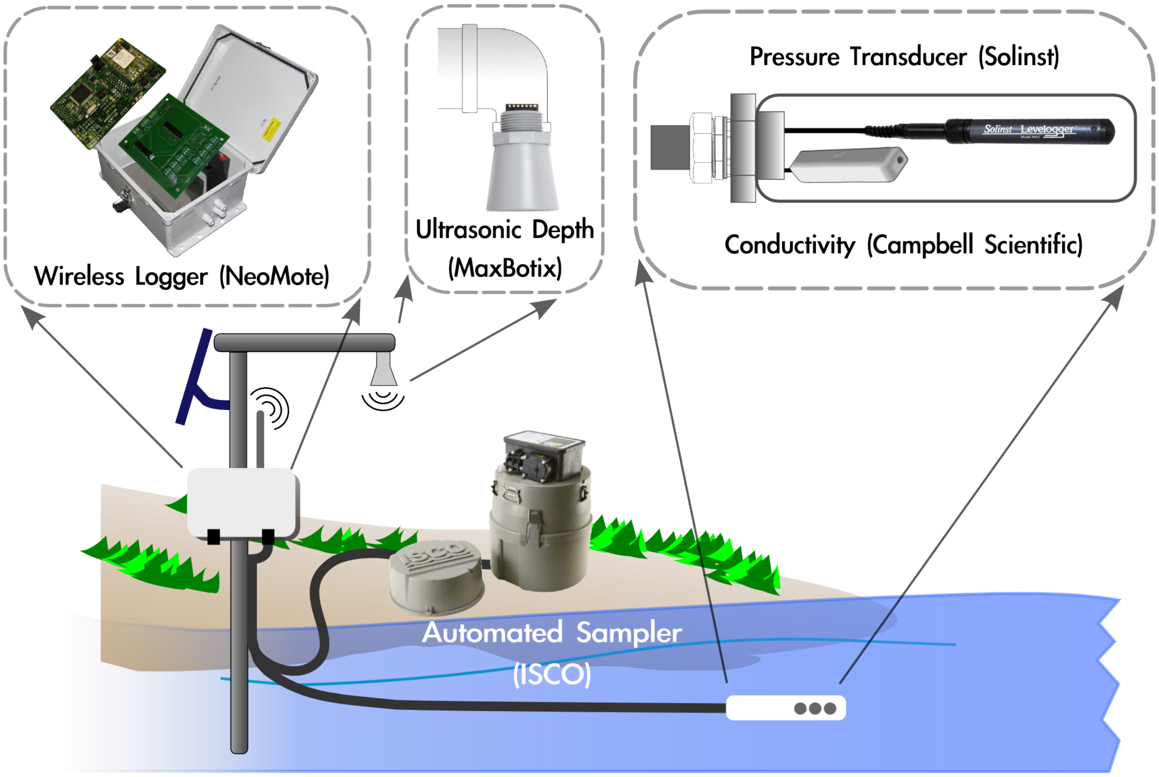

Wireless Logger (NeoMote)
Ultrasonic Depth (MaxBotix)
Pressure Transducer (Solinst)
Conductivity (Campbell Scientific)
Automated Sampler (ISCO)

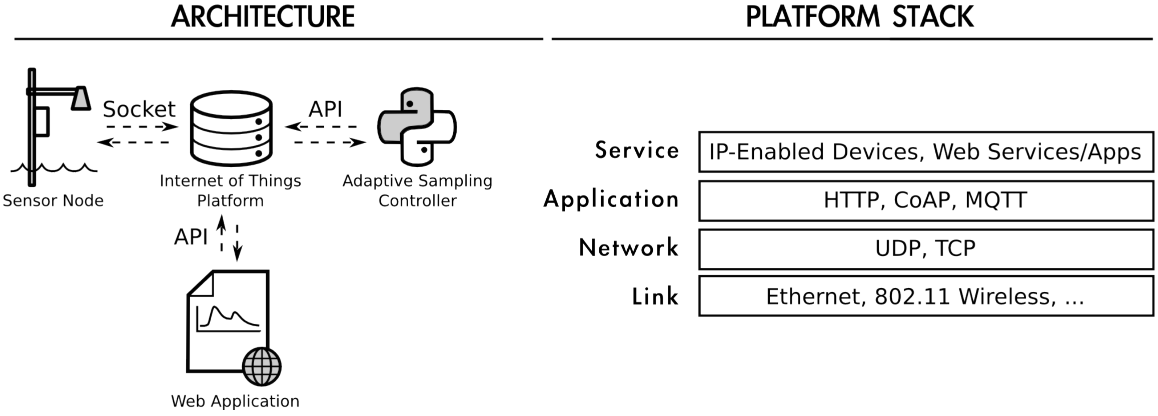
Figure 3.1. IP- enabled hydrologic sensor node which consists of a wireless logger, automated sampler, and a suite of sensors to measure depth and conductivity.

3.3.4 Software architecture

To enable rapid deployment and reliable and secure operations, we designed and implemented a real-time data architecture around an Internet of Things platform. At the time this study was done, there were a number of IoT platforms known to the authors (Table 3.1). During the platform selection process, the features that were considered included authentication and security mechanisms, data storage, throughput limitations, device management interfaces, and available libraries or APIs. The choice to build our architecture around the platform offered by *Xively*[7] was driven primarily by the availability of easy-to-use libraries for a diversity of programming languages, as well as the ability to support multiple authentication keys and user privileges. Given the emphasis on web services, the same architecture and features could have also been implemented using other IoT platforms with some minor payload syntax modifications.

At the lowest level, the platform assumes that all devices and applications are IP-enabled, whether through wired (Ethernet) or wireless (Wi-Fi, cellular, etc.) interfaces. Data transfer can take place either via TCP or UDP protocols, which was chosen based on application-specific performance requirements. While data can be exchanged in a raw format through these low-level socket connections, a number of application-layer protocols, including HTTP, HTTPS and other low-power protocols, such as CoAP and MQTT, are supported to permit the system to interface with popular tools and programming languages. This significantly reduced implementation overhead while simultaneously providing a large support infrastructure in the form of a broad user community, and thus enabled us to focus on the implementation of adaptive sampling rather than the details of low-level data transfer.

[7] https://xively.com



Service	IP-Enabled Devices, Web Services/Apps
Application	HTTP, CoAP, MQTT
Network	UDP, TCP
Link	Ethernet, 802.11 Wireless, ...

Figure 3.2. Web service based real-time data architecture and platform stack.

To illustrate the flexibility afforded by web services, our architecture (Figure 3.2) implemented three separate web services on three separate devices, each of which was programmed in a different language. Each service was implemented using Xively's RESTful API and was written in a programming language most suitable to its purpose and ease of implementation. The first web service was written in C and executed on the sensor node to transmit data and receive sampling commands via a cellular data connection. The second web service was the adaptive-sampling algorithm (controller), whose logic was controlled by a Python script, which could be executed on a local machine or web server to send sampling commands via a RESTful interface in response to the real-time sensor measurements. A third, client-side service was implemented in JavaScript and used a RESTful interface to interface with the IoT platform. It provided access to historical data and allowed a user to issue commands to the sensor node via a website. Each service was individually authenticated and interacted through the IoT platform, which was hosted by a commercial service. The code for this entire reference implementation is available on our public repository[8].

The IoT platform not only interfaced each web service, but also enabled a suite of data discovery and management mechanisms. This allowed any number of authenticated web services to query the system for historical data and metadata, register for alerts, and to obtain direct links to real-time feeds. Organization, exchange and storage of data interacted through a hierarchical structure that is applicable to most types of data streams from any number of devices (Figure 3.3). Each field device (sensor node) was assigned a unique data *feed*, which was further subdivided into individual *streams* (individual sensors and actuators). Users accessed individual feeds or groups of feeds based on their privileges and authentication keys. For example, some users could only view sensor data and metadata, while others can also control sensors or actuators.

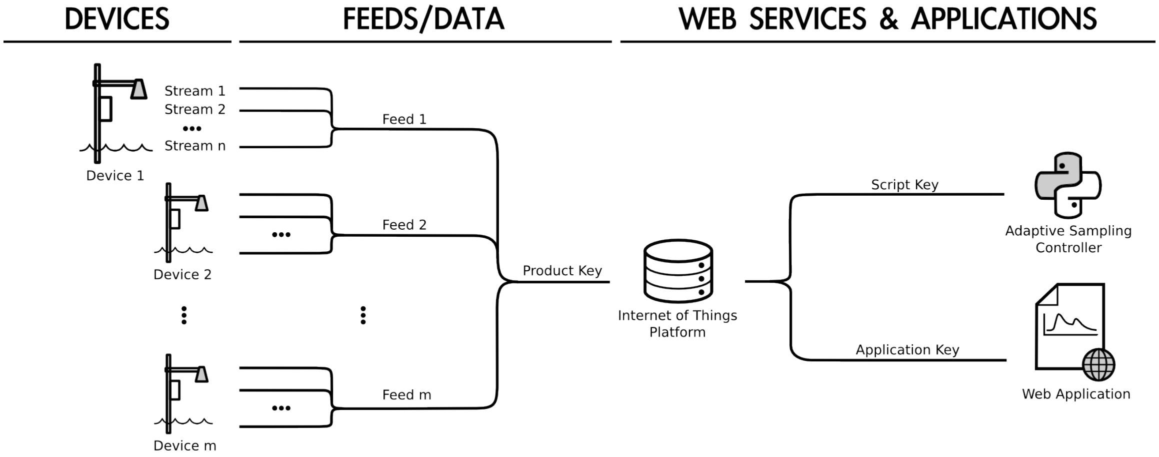


Figure 3.3. Data transport, discovery and storage. All data and services interact through the IoT Platform. Each service is assigned a separate authentication key based on its role and privileges.

[8] https://github.com/kLabUM/IoT

Data and commands were embedded in this structure and exchanged between the individual web service applications using CSV (sensor node) and JSON encodings (sampling controller and web browser application). While XML encodings were also an option, JSON was chosen due to its smaller packet size to reduce cellular transmissions and increase battery life. Given the popularity of these encoding formats across Internet services [81, 82], powerful libraries now exist for almost any programming language to simplify the conversion of sensor readings to formatted data packets, further reducing programming-related overhead. While our application did not explicitly demand the use of domain-compliant syntaxes (e.g. *WaterML, SensorML, DelftFEWS*, etc.*)*, this feature can be added as a relatively lightweight web service or library that maps between the desired syntax formats to drive the adoption of the proposed architecture by domain-specific communities.

3.4 System implementation

Our use case architecture was implemented in three web service modules: the embedded sensor node (programmed in C), the adaptive sampling controller (programmed in Python), and a front-end visualization and control interface (programmed in JavaScript). All services were tied together via the Xively IoT platform, which served as the interface and data storage mechanism.

3.4.1 Sensor node

The majority of the time, the sensor node operated autonomously at a constant measurement frequency, sampling the suite of sensors and transmitting data via the cellular connection. The sensor node's sampling frequency or sampling schedule could be changed remotely via an IoT web service request by authorized users, in particular by the adaptive sampling controller. To control any additional sensor nodes, a user only had to know the unique data feed and authentication key assigned to each device.

Given the rising popularity of ultra-low-power micro-controllers [88], including ARM- [87], AVR- [74] and 8051-based architectures [52], C continues to be the de-facto programming language for the majority of embedded devices. The exchange between the sensor node and the IoT platform was also programmed in C and the Xively platform offers a comprehensive C library, which includes the methods that provide the additional functionality of RESTful communications and data formatting to exchange information with the platform. A number of older or popular data loggers (such as those made by Campbell Scientific[9]) are written in proprietary or legacy languages, for which there may not be an explicit IoT library. Nonetheless, these loggers still support TCP/IP functionality via a number of communicating links. To that end, we decided to forgo the existing Xively library to illustrate the steps that could be followed to interface most IP-enabled data loggers to the IoT platform. To transmit data, our code opened a TCP/IP port and wrote a CSV or JSON-delimited set of sensor values using a RESTful command to the IoT platform. To receive commands from the IoT platform, the node listened on a given port and parsed an incoming string for relevant commands. On our C-based platform, this was achieved in as little as four lines of code by leveraging an existing TCP/IP library.

[9] http://www.campbellsci.com/

3.4.2 Adaptive sampling controller

Implemented as a Python script, this controller sought to maximize the probability of capturing first flush events while minimizing the number of water quality samples. The controller persistently updated its knowledge of local weather forecasts by leveraging *Weather Underground*'s *Weather API web service*[10]. Once notified of the most recent sensor node measurements through Xively, the controller then updated the node's sampling frequency (a variable stored on Xively) based on anticipated storm events.

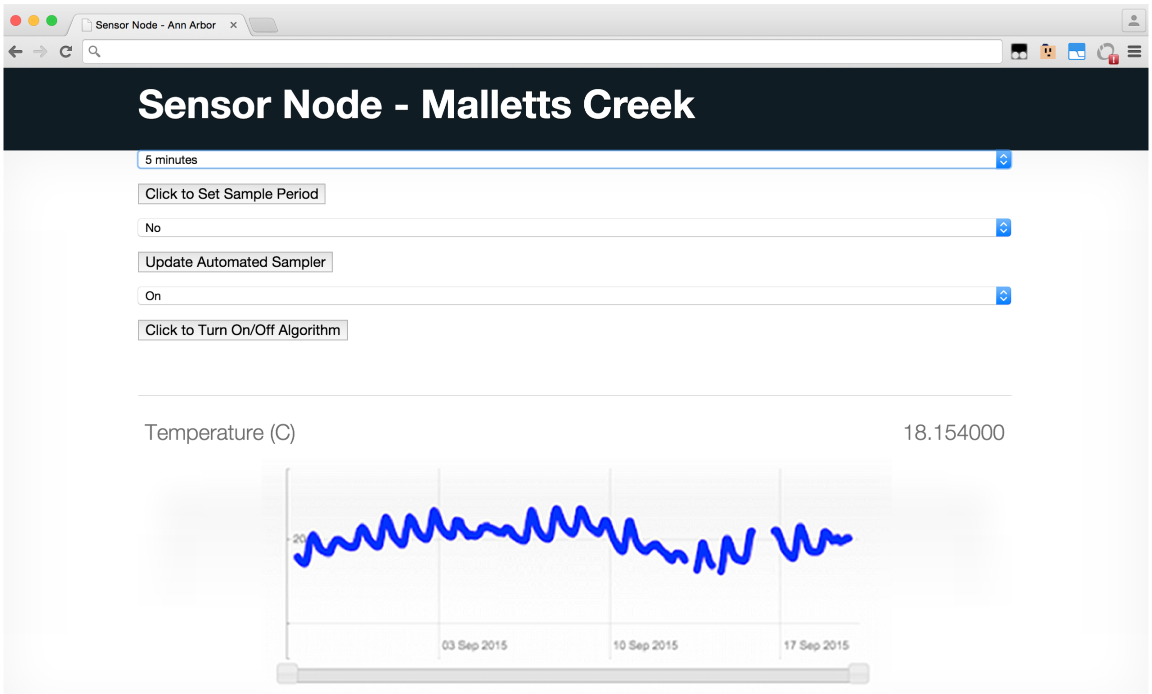
The algorithm was initially developed and tested on an Internet-connected desktop, after which it was deployed as a dedicated web application using Amazon's cloud-based *Elastic Beanstalk*[11] service. This service permits non-expert users to develop code on their own workstation and launch it as a web service by simply uploading the script to the platform. The process involves no further programming beyond what is already written on the desktop computer, which makes it appealing to users who do not wish to support their own dedicated server. The service self-balances computational loads, is pre-configured to support a variety of programming languages, and removes any hosting requirements on the part of the user. Not unique to AWS, a number of other cloud-based platforms offer similar services, including, but not limited to AppFog, CloudBees, Google App Engine, Engine Yard, Heroku, OpenShift, and Windows Azure[12].

[10] http://www.wunderground.com/weather/api
[11] http://aws.amazon.com/elasticbeanstalk
[12] https://www.appfog.com/, http://www.cloudbees.com/, https://appengine.google.com/, https://www.engineyard.com/, https://www.heroku.com/, https://www.openshift.com/, http://azure.microsoft.com/en-us/.

While the adaptive sampling controller could have also been implemented on the actual sensor node in C, we envision future applications where sampling frequencies are guided by measurements made by a distributed network of sensor nodes, rather than just local measurements. In such cases an off-site sampling controller is not only more easily maintained and deployed, but also capable of coordinating a global response to signals from multiple sources. Furthermore, implementing the sampling logic in Python permitted the sampling logic to be updated rapidly (and remotely) through a web interface without having to update the lower level firmware of field devices.

3.4.3 Visualization interface

A front-end web application (Figure 3.4) was implemented as a webpage using the Xively JavaScript API to visualize data and system states. The API provided methods to authenticate with the Xively platform and exchange data in JSON format, which is widely compatible with popular visualization platforms (*d3.js*; for examples, see [96]). Data and commands were transmitted directly to Xively, while a subscription feature in the API enabled callbacks whenever new readings were received from the sensor node. As such, data on the interface were visually updated as soon as they were received by Xively, without requiring the user to refresh the page. A set of controls also allowed users to trigger the sensor node remotely, permitting them to override the logic of the automated sampler if needed.



Sensor Node - Ann Arbor

Sensor Node - Malletts Creek

5 minutes

No

On

Temperature (C) 18.154000

Figure 3.4. Example web front-end used to monitor readings from the sensor node and control its sampling frequency.

3.4.4 Web service interactions

The sensor node was programmed to spend the majority of time in a *sleep state*, where cellular and sensing capabilities were turned off to conserve battery resources, which was required to enable long-term, battery-powered deployments in remote areas. Upon transmitting a new sample, the sensor node remained connected to the Internet for a short duration, giving external services enough time to respond to the new measurements if needed. When in the sleep state, the node did not immediately respond to commands sent by the adaptive sampling controller or the web interface. Rather, it checked for the need to update its sampling schedule once it obtained an Internet connection during its next wakeup cycle. This flow of actions also removed the burden on the adaptive sampling controller to monitor the connectivity of the node, which allowed both processes to remain uncoupled.

A typical set of actions (Figure 3.5) involved a sensor node taking readings and transmitting them to the IoT platform, which then pushed a notification to the adaptive sampling algorithm and web visualization interface, both of which were subscribed to the data feed via their respective library callback mechanisms. The adaptive sampling algorithm then computed the optimal sampling frequency and updated it if necessary, in which case the sensor node was notified via a push notification through the IoT platform. If the sensor node was in a sleep state when the readings had to be updated (for example, due to an unforeseen storm forecast), the adaptive controller updated the sampling frequency on the IoT platform. Upon regaining Internet connectivity, the sensor node could then compare this variable with its current settings and update itself if needed.

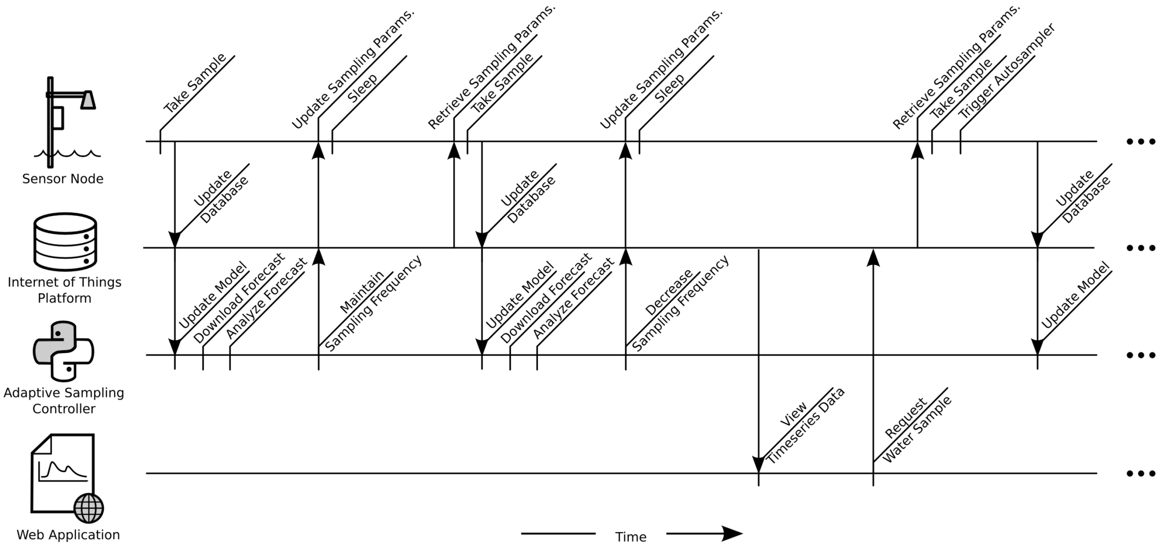


Figure 3.5. Hydrologic use case action flow.

3.5 System performance and discussion

Results from the use case indicate that the sensor node, when guided by the off-site, real-time adaptive sampling controller, resolved local hydrographs while simultaneously collecting water quality samples during events of interest (Figure 3.6). In particular, the node was very effective at managing the number of water quality samples required to characterize the “first flush” behavior of the study basin. Specifically, the node captured valuable baseflow samples right before the onset of a storm, while spacing out the remaining samples to measure water quality during the inflection points, peak, and recession of the hydrograph. At least six samples were used to characterize the dynamics of each distinct rain episode. In many instances, only a single set of samples (no more than 24) was necessary to capture storm events across multiple days without the need to service the node or replace sampling bottles. This would not have been feasible without a real-time sampling approach and has vastly improved the quality of our existing experiments.

During the entire three-month study period, only two baseflow samples were triggered falsely as a result of inaccurate weather forecasts, suggesting that publicly available weather feeds may have high potential to improve urban water experiments. The samples were analyzed for total suspended solids (TSS) to assess impacts of upstream stormwater runoff, showing that peak solids correlated with peak flows. An in-depth water quality analysis, described in [94], concluded that no first flush behavior was observed in the catchment with regard to TSS, which may have significant implications to existing stormwater management practices in the basin. It should be noted that the analysis of a number of other water quality parameters has to conform to maximum holding times (e.g. 24-hour maximum for phosphate), as the constituents may react while the sample is held in the bottle. In these instances, the benefit of real-time data is twofold: permitting for alerts to be sent when samples have to be picked up, and secondly minimizing the number of samples that need to be analyzed in the laboratory.

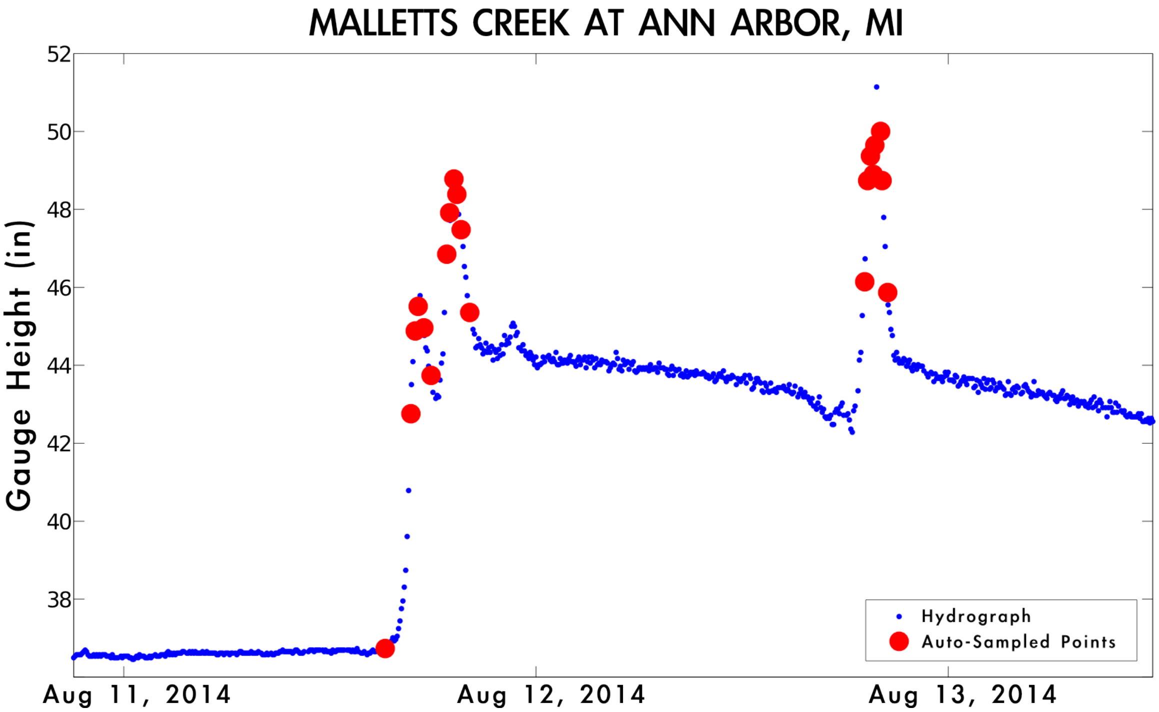


Figure 3.6. Measured hydrologic signal: stage height (blue) and instances of adaptively sampled water quality data (red).

Overall, adaptive sampling significantly reduced fieldwork and improved the power consumption of the sensor node when compared to traditional sampling approaches. Compared to these approaches, which may take water samples once an hour (or more rapidly) during a storm to resolve relevant features of the hydrograph, our implementation was able to more effectively and more densely collect samples storm events. This reduced power draw by nearly 75% since the automated sampler was the largest battery drain due to its mechanical components. The system also provided a number of alerts to users, such as when sample bottles were about to expire, which limited site visits significantly. This reduced requirements on manpower and freed up experimental resources, permitting a multi-node network to be effectively maintained by a small team of investigators. Adaptively sampling these and other signals, however, was highly contingent upon responding to in-situ measurements and weather forecasts in real-time, which required a real-time architecture, such as the one proposed here.

By leveraging a commercial cloud-hosted IoT cloud platform, our real-time use case yielded significant benefits. Using a free account with a cloud-hosted IoT platform, we were able to make our real-time data available on the Internet through a password-protected, web-accessible endpoint, as well as interface our sensor node with a cloud-hosted adaptive sampling algorithm and web application. Development was focused largely on the sensor node and adaptive sampling algorithm. Particularly, no IoT platform outages were experienced throughout the three-month summer sampling campaign (July 1 to October 1, 2014) due to guaranteed uptimes by the platform provider. Building the architecture around a commercial platform also ensured that the overall system would benefit from security and system updates at no expense to the user. The IoT platform also served as an effective data storage, retrieval and visualization engine for continuous sensor streams. As such, the experiment was afforded the benefits of a conventional, non real-time platform as well.

The majority of our use case efforts focused on implementing modular web services, each of which was written in a different programming language and deployed on a system deemed most suitable for its use. The interoperability of web services provided significant flexibility during development and deployment, as it permitted us to focus on the application and leverage our core competencies rather than having to conform to specific languages, operating systems and hardware architectures. Moreover, adjustments could be made to individual web services without affecting or compromising the functionality of the rest of the system. For example, implementing our Python-based adaptive sampling controller as a web service allowed us to rapidly change sampling strategies without having to modify any low-level firmware on the sensor node.

In the example use case, all of the web services interacted through the IoT platform. In such an architecture, feeds and alerts must be routed rapidly enough to meet the needs of the application. Latency thus becomes a concern when framing any architecture around an IoT platform. To address this concern, we carried out an experiment in which two web services were created and connected via the Xively IoT platform. The controlled experiment was designed to emulate an adaptive sampling procedure where a sensor node first transmits a reading that is interpreted by an off-site controller, which then instructs the sensor node of its new sampling schedule. Three hundred data packets were transmitted from one web service, forwarded by Xively to the second service, and then transmitted back through Xively to the first service. The total travel time was measured, yielding an average of 0.2 seconds round-trip (min 0.002 seconds, max 2 seconds). This overall latency was not only guided by the response time of IoT platform, but by other factors such as network connectivity and bandwidth. In our hydrologic use case, where the average sampling interval was rarely required to drop below five minutes, this response time was more than adequate to meet the needs of the application. Such a performance should also adequately meet the needs of the vast majority of real-time environmental applications, most of which rarely require sub-second temporal resolutions for purposes of control and decision-making. Applications requiring very fine-grained response times (milli- to micro-seconds) can still leverage the majority of features offered by IoT platforms but should consider more localized, on-board signal processing and control where possible (e.g. on the sensor node, as opposed to off-site services). In the case where an IoT platform still does not meet the needs of such applications, it can still serve as a directory and discovery mechanism that interfaces web services, which can then communicate with each other directly. Such customized architectures should, however, rarely be required for the vast majority of environmental sensor network applications.

Aside from latency concerns, a web-service architecture could suffer from connectivity outages as well. As pointed out by [62], the overall functionality of a web-coupled architecture may suffer if it is entirely reliant upon being interconnected by the Internet, especially when services are hosted at different locations. Connectivity outages (for example, those experienced in wireless sensor network applications) may thus intermittently affect portions of a real-time architecture. To that end, a level of autonomy should always be built into individual services to ensure that they maintain their core set of functionalities even if connectivity is compromised. In our use case, the sensor node would continue to sample and transmit data at a default sampling interval, even if the adaptive sampling controller were to experience an outage. On-board storage on the sensor node also maintained a local copy of the data that could then be re-uploaded in the future, ensuring a continuous stream of data regardless of IoT functionality.

In most real-time environmental applications, particularly those relating to smaller-scale scientific studies, commercial platforms will often provide low-cost or free operations, with minimal overhead to setup and begin using real-time services provided by the platform. However, the management and control of commercial IoT platforms is subject to provider policies and subject to future changes. As is the case with most commercial software and systems (e.g. cloud computing services), potential drawbacks include throttled usage and loss of support for certain features. The benefits offered by open source platforms make them a viable real-time alternative for users willing to commit resources to both developing maintaining the platform. While they require more expertise during setup and maintenance, open source platforms may be a viable option for more advanced users already experienced in developing and deploying web applications as they allow more control over their system and their data. These platforms provide more low-level configurability to the user and collected data does not have to reside on third party databases. Furthermore, users are not explicitly restricted by usage limitations or by data formats. Nonetheless, when considering the use of IoT platforms and web services for real-time environmental applications, users will also ultimately need to weigh the benefits of and drawbacks of commercial platforms against their open source counterparts.

3.6 Conclusions

Recent advances in sensing, computation and communications have enabled the rapid deployment of real-time data systems for environmental applications. In particular, most modern sensor systems can now seamlessly connect to the Internet via standard web protocols, permitting the use of web services as an ideal interoperability mechanism between sensors, actuators, models and decision support systems.

The ability to respond to data as it is measured brings two major benefits to environmental applications: 1) it enables a means by which to significantly improve the quality and reach of experiments (as illustrated by our hydrologic use case), and 2) it serves as a powerful tool for decision-making and control (e.g. contaminant warning systems, flood control, etc.). Even with the current ecosystem of open source platforms, the deployment of current real-time environmental data systems is largely non-trivial, which significantly limits their adoption. To that end, we have shown that commercially available IoT platforms, which have been designed for a broad suite of applications, provide a secure and scalable mechanism for processing, storing, and visualizing ever increasing amounts of data.

The flexibility afforded by the web service-driven nature of these platforms loosens the architecture-, hardware- and software-specific requirements that often underpin several existing real-time data platforms. As illustrated by our hydrologic use case, this flexibility reduces the barrier of entry for most environmental applications as it permits users, novice and experienced, to build upon existing projects and work with the programming languages and platforms that they find most appealing. Regardless of the ease of use, demands on the user are not entirely eliminated. An initial investment to develop the appropriate skillset to work with IoT platforms will inevitably have to be made by end users. We contend, however, that learning how to integrate web services into already existing code provides a compelling value proposition given the large community of adopters and supporters that is growing in the IoT space.

While the availability and features of open source IoT platforms continue to expand, environmental applications presently stand to gain the most from leveraging commercial IoT systems, which offer a vast suite of features at the click of a mouse. Presently, these enterprise-quality platforms have the potential to enable the ubiquitous use of real-time environmental sensor data and to usher in a new generation of adaptive scientific experiments.

Chapter 4 Adaptive measurements of urban runoff quality

4.2 Introduction

The ability to seamlessly measure urban watersheds, which was enabled by the technological outputs of the prior chapter, now enables highly tailored experiments to be carried out to determine unique water quality dynamics that may be inherent in a given watershed. In turn, this will permit control of infrastructure to be finely tuned to watershed-specific outcomes. Nonpoint source pollution is a leading cause of surface water impairment in the United States and represents a major management concern as rapid urbanization continues to strain local and regional water resources [97, 98]. The emergence of reliable environmental sensors is poised to transform our understanding of nonpoint source pollution and broader water systems [99, 100]. In hydrologic studies, new sensors are revealing previously unmeasured dynamics that govern water quality across large watersheds. For example, new optical nitrate sensors are improving the quantification of loads, flow paths, and nutrient dynamics [101-103]. Furthermore, the recent ability to continuously measure turbidity and sediments has challenged existing assumptions of sediment variability, suggesting that nutrient concentrations exhibit complex dynamics that often cannot be attributed to storm features [104].

While these sensor measurements will help to fill critical scientific knowledge gaps, the management of water systems also stands to significantly benefit from an improved understanding of water quality dynamics. Much of urban water quality management is tuned to handle the storm as a whole, seeking to control and treat the cumulative event rather than affect its dynamics. This is accomplished through a variety of *green* or gray infrastructure solutions [105, 106], the choice of which is often based on assumptions of stationarity and few or no measurements. While improved measurements of water quality will help to guide the design and maintenance of these systems, a new generation of *intelligent* infrastructure (controllable ponds, tanks, weirs, bioswales, etc.) stands to benefit even more from improved quantification of pollutant dynamics. Modern infrastructure will soon route water in real-time to respond to individual storm events [45, 107-109] to reduce flooding and improve water quality. Such finely grained control will benefit from an equally finely grained understanding of water quality dynamics.

However, the widespread use of in-situ sensors is still limited by costs, high power consumption, and maintenance requirements. Moreover, for many important parameters, such as metals, there are no in-situ sensors to provide such measurements. For emerging contaminants, including viruses and industrial chemicals, in-situ sensors may never become available unless regulations or research drive their development. Automated samplers, which retrieve water samples for subsequent laboratory analysis, may be used to fill these measurement gaps. While they may incur considerable expense for installation, maintenance, and repair [110], automated samplers provide a flexible and automated means by which to reduce man-hours that would otherwise be required to achieve the same task.

Advances in wireless communications and data architectures are now significantly reducing the overhead required to deploy environmental sensor networks [111-113], enabling the adaptive and real-time study of water systems. These advances are however not being leveraged to their maximum potential [114, 115], as the majority of presently deployed sensor platforms are still used in an *off-line* fashion. By adapting a study to in-situ conditions and various public sources of real-time data such as weather forecasts or streamflow measurements, the quality of the final experiment stands to significantly improve. This is particularly pertinent in the study of hydrologic systems and nonpoint source water quality, where abrupt changes in water quality due to unanticipated flashy storms often contain critical information about water quality dynamics in watersheds [116].

The goal of this chapter is to investigate a scalable approach by which to adaptively measure nonpoint source water quality in urban watersheds with the specific objective of characterizing dynamics (timing and magnitude) of pollutant runoff. An adaptive sampling algorithm is introduced, which executes on sensor nodes and queries local weather forecasts to anticipate state changes in a hydrograph signal. These state changes are then used to guide an online sampling schedule to minimize the resource consumption of a sensor node, while simultaneously maximizing the information content of the acquired water quality measurements. The adaptive sampling method is evaluated during the 2014 rain season to study the dynamics and *first flush* behavior of total phosphorus and total suspended solids (TSS) in an urbanized watershed. While urban nonpoint source water quality is the focus of this chapter, the methods presented herein can readily be adapted to a broad suite of other resource-constrained hydrologic and water quality studies.

4.3 Background

4.3.2 Problem description

The study and management of watersheds and drainage networks often hinges upon an accurate detection and characterization of transient events, as the remainder of the system is often in a steady, relatively well-understood state. For many urban hydraulic and hydrologic systems these rapid changes are driven by highly uncertain phenomena, such as precipitation [117, 118]. Knowledge of water quality dynamics during storm events provides a guiding principle for nonpoint source urban water quality control, which has most recently been brought to the public's attention through the meteoric rise of *green infrastructure*, particularly across much of the United States [119]. Beyond green infrastructure, many cities also implement a variety of *Best Management Practices* [120], several of which are designed to route initial flows toward large retention or detention basins for settlement or infiltration. In the American Midwest, some of the most critical water quality measurements include nutrients, particularly runoff-generated phosphorus. While algal blooms and eutrophication are driven by complex dynamics that require both nitrogen and phosphorus, in many freshwater systems, such as the Great Lakes, phosphorous is often the limiting nutrient [121, 122].

A large body of research has shown that runoff pollutant concentrations exhibit highly complex dynamics that depend, among many other factors, on the type of pollutant, intensity of rain events, the physiography of watersheds, local flow regimes, and antecedent dry periods [117, 118, 123-125]. One popular concept in urban hydrologic research is the "*first flush*" of pollutants into streams and rivers [104, 126, 127]. This effect has been known to occur particularly in urban streams that display leading *hysteresis*, where the highest concentration of contaminants occurs at the beginning of a storm event, as contaminants are first washed off roads and other impervious surfaces. However, a number of studies have not observed the first flush [104, 128-130], showing that peak pollutant concentrations do not always arrive within a small fraction of the initial runoff (Figure 4.1). While the first flush is an important phenomenon, this initial fraction of runoff may not be the primary or only source of pollutant loadings for some watersheds and chemical constituents. In some streams, high levels of erosion caused by local flow regimes that exceed geomorphically significant levels are a leading cause of suspended sediment and nutrient loads [22]. For such streams, peak loads of sediments are often correlated to flows rather than a first flush. To that end, there is a need to collect representative measurements of storm-driven water quality dynamics to improve our fundamental understanding of land-use practices on water quality.

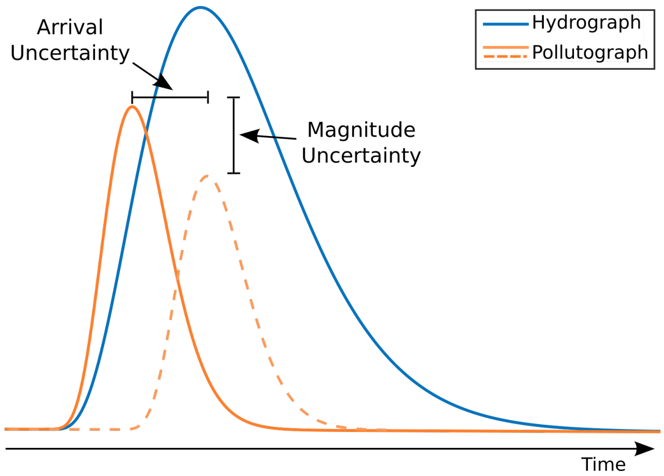


Figure 4.1. Characterizing the peak pollutant runoff arrival is critical to informing impacts of land use and water quality control practices, which often rely on assumptions of a *first flush*.

4.3.3 Instrumentation

When compared to water flow, water quality remains relatively expensive and difficult to measure [116]. Even today, despite advances in telemetry and low-power microcontrollers, a dense spatial coverage of in-situ water quality measurements is still hindered by a lack of reliable and cost-effective sensors. For many important parameters, such as nitrate, the cost and power consumption of sensors inhibit their ubiquitous deployment, while for other parameters, such as phosphorus and phosphates, non-colorimetric or in-situ sensing technologies do not even exist [131]. Many water quality sensors also consume more energy than the entire remainder of the data acquisition system and require frequent servicing to mitigate field effects such as biofouling [101, 132]. As such, maintaining sensor networks to measure water quality across large geographic areas is a resource-intensive task that presently poses a major barrier to the ubiquitous measurements of urban water quality.

When continuous in-situ sensing becomes too expensive or infeasible, field-hardened automated samplers can be used to collect samples, which can be subsequently analyzed in the laboratory for a variety of water quality parameters (e.g. nutrients, metals, solids, bacteria, and other emerging contaminants) [89, 110, 116, 129]. These units are programmed to pump a sample of water into one of a number of bottles. Depending on the study objectives, these samples usually range from one 20-liter bottle to as many as 24 one-liter bottles. The use of automated samplers presents a set of unique deployment challenges compared to in-situ sensors. In the absence of grid power, the significant mechanical energy required to physically pump samples places a major drain on battery resources. Additionally, samples may need to be refrigerated or chemically treated for preservation depending on the constituents of interest [133, 134]. As is the case in the use of most other sensors, autosamplers are also plagued by the need to calibrate readings to variability in a stream cross-section. For dissolved constituents, selecting a well-mixed site can remedy this variability as a sample at a single point may then be assumed to be representative of the entire stream cross-section.

While one-bottle samplers are a practical means by which to study the composite effects of a storm event, they do not provide insight into the detailed dynamics of an event, which is important if they are to be used as substitutes for continuous, in-situ sensors. When using multiple sample bottles to resolve urban pollutograph dynamics, the limited number of available bottles becomes a major constraint. If the timing, magnitude, and duration of storms are not accurately anticipated, '*wasted'* or missed samples often become a common experimental occurrence. Measuring too slowly can entirely miss the dynamics of an underlying pollutograph (Figure 4.2a). On the other hand, measuring too fast or too early may deplete the number of sample bottles before an event is fully captured, which is particularly common if storms last for multiple days (Figure 4.2b). To mitigate this, units can be configured to acquire samples if a pre-set flow threshold is exceeded, after which the hydrograph can be sampled according to predetermined flow- or time-weighted intervals [89, 135]. However, this strategy may miss important baseflow samples. Also, as storm duration and intensity can be highly variable, setting triggers or intervals to static values may not consistently sample a wide range of storm events. Flow-weighted sampling cannot account for storm intensities that deviate far away from the design storm or have multiple distinct discharge peaks (Figure 4.2c). Furthermore, the number of available bottles may still be depleted before an event is fully captured if the storm lasts longer than expected. While missed baseflow concentrations can sometimes be estimated from samples taken during other dry weather periods [136, 137], such estimates may be inaccurate since elevated concentrations may occur at the onset of a storm [138]. None of the conventional sampling techniques distinguish between important points of the flow hydrograph, such as the peak and inflection points, which may often contain significant information with regard to the effect of land-use variability on the pollutograph.

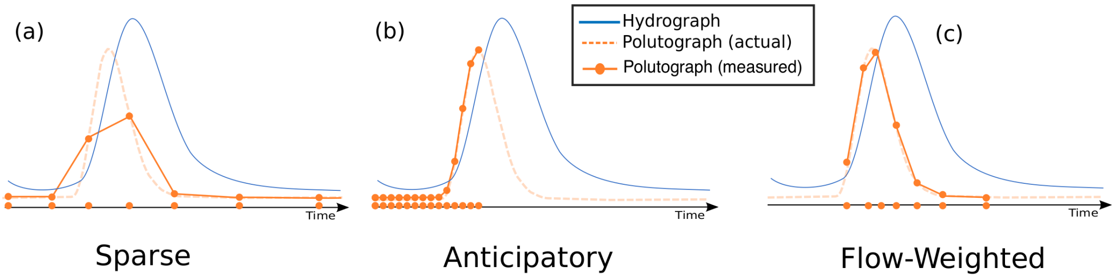


Figure 4.2. (a) Undersampling reduces the use of constrained experimental resources but can lead to an improperly reconstructed water quality signal, (b) rapid sampling may deplete all of the sampling resources before the event is complete, (c) triggering on flow thresholds or storm intensities alone may miss the onset of smaller events.

4.3.4 Adaptive sampling

The concept of adapting measurement strategies or detecting events of interest has been introduced broadly in the signal processing and machine learning literature for a variety of applications but has seen limited use in hydrology. Often, *adaptive sampling* revolves around spatial measurement strategies, where measurements at one location are used to inform locations of new measurements [139]. The problem can also be extended to the temporal domain, where sampling frequencies are changed during events of interest [140]. The task of detecting these events falls broadly into the literature of *change-point detection* [141, 142], where a signal is monitored to isolate abrupt state changes or transient events. While few studies couple these two objectives, in the case of water quality, *adaptive sampling* and *change-point detection* are inherently coupled, as the detection of a hydrograph change must be accompanied by a change in the sampling schedule to resolve the features of the pollutograph. Much of the existing literature on these topics does not explicitly incorporate the physical dynamics or nuances of such phenomenon, which limits their benefit to many real-world experiments.

While automated samplers provide a way to sample many water quality parameters simultaneously, the off-line use of these devices impedes their scalability as an experimental platform. The use of in-situ measurements coupled with real-time data, which is readily afforded by current technologies, has the potential to transform these sampling strategies from static to highly adaptive. For example, [89] reprogrammed an automated sampler to distribute 20 sampling bottles throughout a storm event. This approach, however, did not consider explicit hydrograph states or weather forecasts, which may cause valuable measurements to be missed. To that end, real-time data processing and adaptive sampling will allow sensing resources to be continuously optimized around site-specific conditions to ensure that measurements are taken at the most informative points.

4.4 Methods

A real-time framework for the adaptive sampling of water quality is presented, which controls automated samplers to minimize the number of sampling bottles required to reconstruct the temporal dynamics of the pollutograph. The method continuously adapts to individual storm events by incorporating real-time weather forecasts and updating a local model of flow conditions to trigger samples in response to hydrograph features rather than predetermined timing or flow thresholds. The technique is designed to be computationally simple enough to be executed efficiently on a field-deployable microcontroller, but can also be readily ported to the *cloud* or remote servers. The approach (Figure 4.3) forms an embedded processing chain, leveraging local and remote computational resources to assimilate real-time sensor measurements into a model of local water flow. The core of the architecture is comprised of embedded, remotely-deployed, and internet-connected sensor nodes, which obtain live meteorological forecasts from public web services to persistently update the probability of precipitation in the study area. Measurements from a local depth sensor are continuously fed to a state estimator, which estimates the flow dynamics of the stream. These estimates are then fused with the latest weather forecast and routed to a sampling controller, which determines when the next sample should be taken by the autosampler.

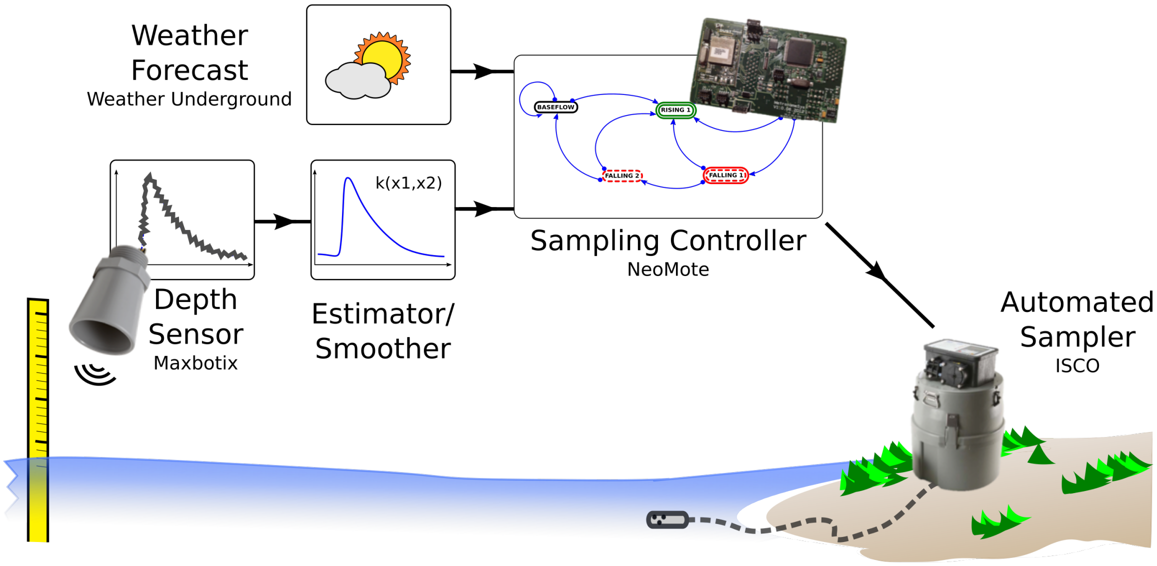


Figure 4.3. Forecast data is acquired in real-time from the Internet and fused with filtered, real-time sensor data to trigger the automated sampler based on relevant hydrologic states.

4.4.2 State estimation

The state of the hydrograph must first be estimated before water quality measurements can be scheduled. Let the state $x(t)$ denote the flow (or stage) of the hydrograph at time t. We assume that the measured flow is corrupted by noise, $\varepsilon(t)$, such that a sensor measurement $y(t)$ is given by (4.1):

$$y(t) = x(t) + \varepsilon(t) \tag{4.1}$$

where $\varepsilon(t) \sim N(0, \sigma^2)$ is normally distributed, zero mean. Given the real-world performance of most sensors, the measurement noise can be taken as stationary, with a variance σ^2 that can readily be obtained from manufacturer datasheets or a simple laboratory evaluation.

In most applications, rather than triggering new samples based on the actual flow, it may be more relevant to trigger samples based on the first or second derivatives of the flow, which are indicators of important hydrograph features independent of storm duration and magnitude. For example, it is often of interest to distinguish between the rising or falling limbs of the hydrograph:

$$\begin{aligned} &\frac{dx}{dt} \geq 0 \quad \text{rising hydrograph limb} \\ &\frac{dx}{dt} < 0 \quad \text{falling hydrograph limb.} \end{aligned} \tag{4.2}$$

The first derivative can be used to detect the onset of a storm event or find the hydrograph peak, while the second derivative of the flow d^2x/dt^2 can be used to detect inflection points, which are indicators of precipitation intensity or baseflow conditions. For notational simplicity, let $\dot{x} =$

dx/dt and $\ddot{x} = d^2x/dt^2$. Given the noise in real-world signals, directly differentiating the noisy signal y would only amplify the effects of the noise, thus obscuring any meaningful estimate of derivatives. Thus, an improved estimate of x must first be obtained in real-time before $\dot{x}$ and $\ddot{x}$ can be used to make sampling decisions. This is particularly true in smaller storms, for which changes in flow may be subtle.

We derive a noise-free estimate $\hat{x}(t)$ through a non-parametric kernel smoother [143]. For a noisy observation y_j at time t_j let $\hat{x}(t_j)$: $\mathbb{R}^n \rightarrow \mathbb{R}$ be a function that obtains a local estimate of x_j through the kernel operation:

$$\hat{x}(t_j) = \hat{x}_j = \frac{\sum_{i=1}^{n} K(t_j, t_i)\, x_i}{\sum_{i=1}^{n} K(t_j, t_i\,)} \qquad (4.3)$$

where $K(\cdot)$ is the kernel function and *n* is the number of observed points to be weighted. Given the normally distributed noise assumption, a good choice of kernel is given by the radial basis function:

$$K(t^*, t_i) = exp\left(-\frac{(t^* - t_i)^2}{2r^2}\right) \qquad (4.4)$$

where r is the length-scale parameter. This kernel smoothing operation weighs the importance of neighboring measurements based on their distance (time, in this case) to the measurement of interest. This smoother is ideally suited for the proposed application, as it does not assume that measurements are taken at even time intervals. Furthermore, this state estimator is very computationally efficient, permitting its implementation on computationally-constrained, low-power microcontroller platforms or data loggers. Once the measured data has been filtered, an estimate of the noise free derivative can be obtained by numerically differentiating the smoothed state.

4.4.3 Adaptive sampling algorithm

Once estimates of x, $\dot{x}$, and $\ddot{x}$ have been obtained, the sampling objective becomes to decide when to take the next measurement. This can be accomplished by scheduling a future sensor reading at time $t + t_s$, or by changing the sensor sampling frequency to $f_s = (t_s)^{-1}$. Often, the sensor used to derive the flow estimates $x(t)$ consumes fewer resources than the sensor used for water quality measurements, as is the case with the automated sampler used in this study. As such, measurements of water height or flow can be made at a higher frequency and used to drive measurements of water quality.

A real-time probability of precipitation, obtained in our case by querying the public *WeatherUnderground* forecast [144], is used to trigger the autosampler to take a water quality sample before a storm. This provides a valuable baseflow measurement and safeguards from missing measurements during instances when the hydrograph changes too rapidly or at too small of an amplitude to be detected by flow sensors alone. The sampling algorithm (Figure 4.4) uses the weather forecast to trigger a sample when the chance of precipitation exceeding 5 mm within the hour surpasses 10% (empirically determined based upon an analysis of historical forecasts and the resulting hydrologic response). Samples are then subsequently triggered based on the estimates of the hydrograph state. While many sampling strategies are possible, in the case of this study, the states of interest included (1) baseflow conditions right before a storm, (2) the onset of the hydrograph to detect a potential first flush, (3) the inflection-point of the rising limb of the hydrograph (4) the peak of the hydrograph, (5) the inflection-point of the falling limb of the hydrograph, and (6) the falling limb of the hydrograph as it returns to within 10% of the prestorm baseflow. In the case that the weather forecast is erroneous, the initiation of a storm event is also marked when the slope in the hydrograph exceeded 7.5 m^3 over 5 minutes, which for our study site corresponded with the minimum observed change in flow from baseflow conditions caused by 5 mm of precipitation in one hour. The algorithm can also be viewed as a state machine, where samples are triggered during state transitions, as determined by estimates of the flow x and its derivatives. The state machine is designed to account for multiple flow regimes (such as delayed surface flows from neighboring slopes), taking additional samples if multiple inflection points or local hydrograph peaks are detected.

Algorithm 1

```
Inputs: Rain Forecast (R), Flow or Stage (x)
  Initialize the current state $s$ to "Baseflow"
  while TRUE do
      Update the state $s$ and sample time $t_s$ based on $(\dot{x}, \ddot{x})$
      if s changed
          Trigger Sample
      else if $R$ and $s$ = "Baseflow" then
          Trigger Sample
      Wait $t_s$ before next update
```

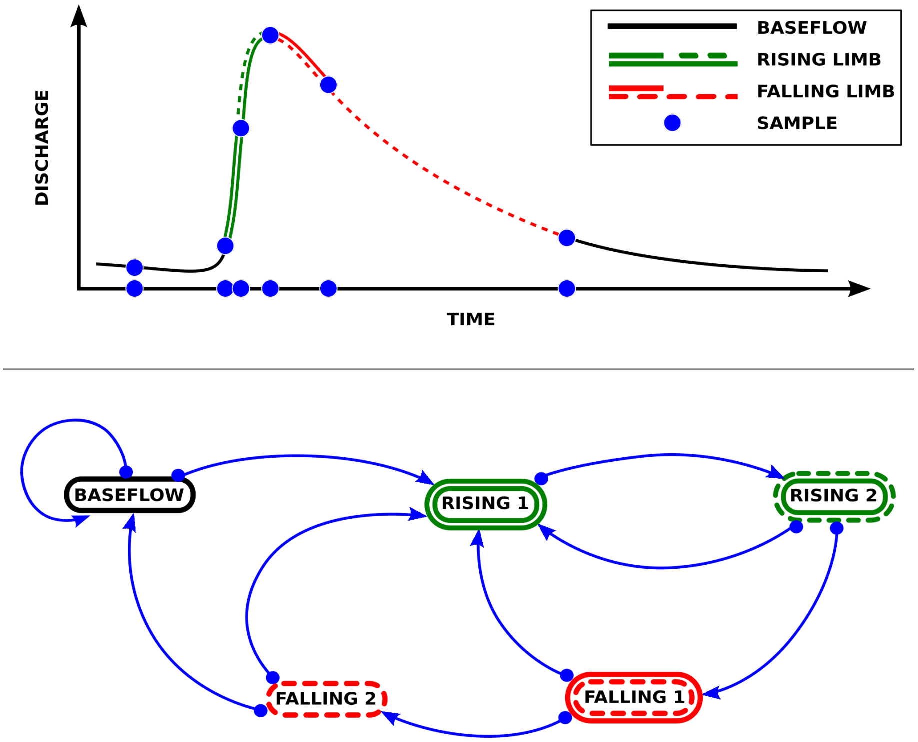


Figure 4.4. Adaptive sampling algorithm (above) and corresponding state machine (below).

4.4.4 Study area, sensors, and cyberinfrastructure

The adaptive sampling algorithm was tested on a sensor node deployed during the 2014 rain season at the outlet of an urban watershed near Ann Arbor, Michigan (Figure 4.5), Latitude 42°15'53", Longitude –83°41'18"). The outlet drains into an end-of-line water quality detention basin located along the Huron River. Ann Arbor's climate is classified as humid continental with severe winters, hot summers, no dry season, and strong seasonality. Annual precipitation is 955 mm and snowfall is 1450 mm. The study area comprises a 28 km^2 catchment that is over 80% impervious with the large concentration of impervious surfaces located near the centroid of the watershed. By the Richards-Baker flashiness index [145], the catchment has a seasonal index of 0.653, which is relatively high for streams in Michigan.

A sensor node and real-time cyberinfrastructure, whose technical details are described in [94, 146], were deployed in the northeastern outlet of the watershed. The sensor node is equipped with a low-power microcontroller (ARM Cortex-M3 architecture) and a low-power wireless module (Telit CC864-DUAL) to take advantage of urban cellular coverage. For the purposes of this experiment, the node was interfaced with a low-cost, low-power ultrasonic depth sensor (MaxBotix MC7384, 3.1mA at 5VDC) to measure the stage of the hydrograph every five minutes, as well as an automated sampler (ISCO 3700, standby: 10 mA at 12VDC, sampling: 2000mA at 12VDC) that drew samples from the *run* of a stream, where channel features were deemed moderate and homogenous [147]. To resolve runoff-driven quality dynamics, a 24-bottle configuration of the automated sampler was used. Weather forecast data was queried every five minutes. For comparison of stage measurements, the node was collocated with a USGS gage (USGS 04174518). Upon validation of the stage estimates, the rating curve of this gage was used to derive flow from our depth readings. In this study, this permitted for flow, rather than stage, to be used to trigger the automated sampler.

4.4.5 Water quality analysis

The samples taken the by the automated sampler were analyzed for total phosphorus and TSS according to EPA Methods 365.3 and 160.2, respectively [133]. EPA Method 365.3 uses a two-step pretreatment and colorimetric approach to determine total phosphorus concentrations while EPA Method 160.2 determines TSS concentrations by first filtering a sample and drying the non-filterable residue in an oven to a constant weight. Bottles were pretreated and collected within twenty-four hours of each storm event to ensure samples were properly preserved prior to analysis. TSS was chosen due to its surrogate relationship with many other contaminants including total phosphorus [148, 149], which was analyzed due to the study site's proximity to Lake Erie, where loadings of total phosphorus are of interest to the study of algal blooms [150].

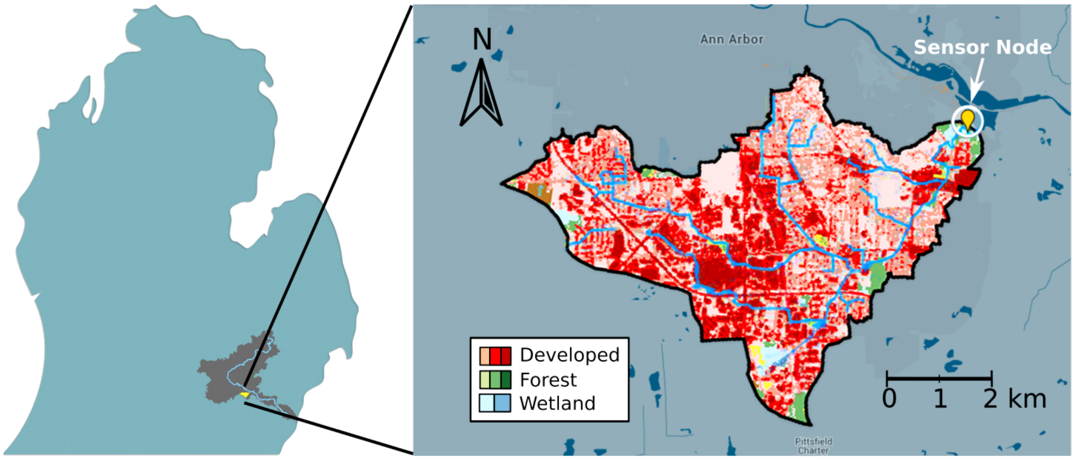


Figure 4.5. The study was conducted in Southeast Michigan at the outlet a 28 square kilometer urban watershed.

To characterize nutrient dynamics and *first flush* behavior, lab results for each storm event were analyzed using *cumulative mass-volume curve* or *M(V) curve analysis* [104, 128], which compares the dimensionless ratio (percentage) of the cumulative flow-weighted concentration with the cumulative runoff over the course of a storm event. This analysis permits the water quality dynamics within multiple storm events to be compared by normalizing for factors such as storm duration or quantities of loading. To identify the existence and strength of a first flush, each M(V) curve was approximated with a power law function:

$$M(k) = V(k)^b$$

$$= \frac{\sum_{i=1}^{k} C_i Q_i \Delta t_i}{\sum_{i=1}^{N} C_i Q_i \Delta t_i} = \left(\frac{\sum_{i=1}^{k} Q_i \Delta t_i}{\sum_{i=1}^{N} Q_i \Delta t_i}\right)^b \tag{4.5}$$

where $M(k)$ and $V(k)$ are the normalized cumulative mass and volume, respectively, up to the kth sample of a given storm event over which N total samples are taken; C_i, Q_i, and Δt_i are the concentration, discharge, and sampling frequency, respectively, of the ith sample [104, 128]. The value of b is inversely proportional to the strength of the first flush (i.e., a value much less than unity, $0 \leq b < 0.185$, would correspond to a strong first flush) and the fit is considered satisfactory for $r^2 > 0.9$ [128]. For each event, the b-value was estimated by minimizing the least-squares fit between equation (5) and the individual data points.

To characterize the variability of pollutant concentrations between storms, the *event mean concentration* (EMC) was also calculated. The EMC normalizes the total event load by the total event runoff volume, yielding a flow-weighted average of the pollutant concentration [104, 117, 127]. It has been shown that in urban environments, peak EMC of pollutants in stormwater runoff can be as much as twenty times larger than baseflow EMC during dry weather conditions [127]. The EMC was used in this study to quantify the constituent concentrations carried by runoff in comparison to baseflow conditions for each storm event. The influence of other factors to event mean concentrations, such as antecedent dry conditions [151] and storm intensity [128], was also considered in the analysis.

4.5 Results

4.5.2 Adaptive sampling algorithm

During the 2014 deployment season, the adaptive sampling algorithm was evaluated on four storm events (July 1, August 11, August 19, and September 10). Each event was preceded by at least a 48-hour antecedent dry period followed by a storm where at least 5 mm of precipitation fell within 24 hours (Table 2). The July 1 event was driven by a 9.4 mm storm over 2 hours with a peak flow 2.78 m^3/s; the August 11 event was driven by a 24 mm storm over 7 hours, characterized by an initial peak flow of 1.30 m^3/s followed 2 hours later by a peak flow of 2.01 m^3/s; the August 19 event was driven by a 9.4 mm storm over 2 hours with peak flow of 4.33 m^3/s; the September 10 event was driven by a 36 mm storm over 5 hours with an initial peak flow of 4.70 m^3/s followed 2 hours later by a peak flow of 5.27 m^3/s.

The state estimator and real-time kernel smoother correctly identified the pertinent flow regimes, triggering the automated sampler within an average of 3.5 minutes (standard deviation $\sigma = 3.8$ minutes) to collect water quality samples as dictated by the control logic (Figure 4.6, example of August 19 event). The relation between the stage y, as estimated by the sensor node, and the discharge Q measured by the nearby USGS gage was found to be $Q = 0.729\ (y - 32.5)^{2.47}$ and was derived using a least-squares fit ($r^2 = 0.993$). The real-time kernel smoothing operation was important to obtaining accurate state estimates, as directly taking the derivative of the sensor signal yielded a noisy, zero-mean signal that could not be used to determine meaningful changes in the hydrograph. Integration of real-time weather forecasts into the control logic ensured that the automated sampler was triggered just before the onset of a storm, allowing for baseflow and background conditions to be decoupled from storm-driven water quality dynamics.

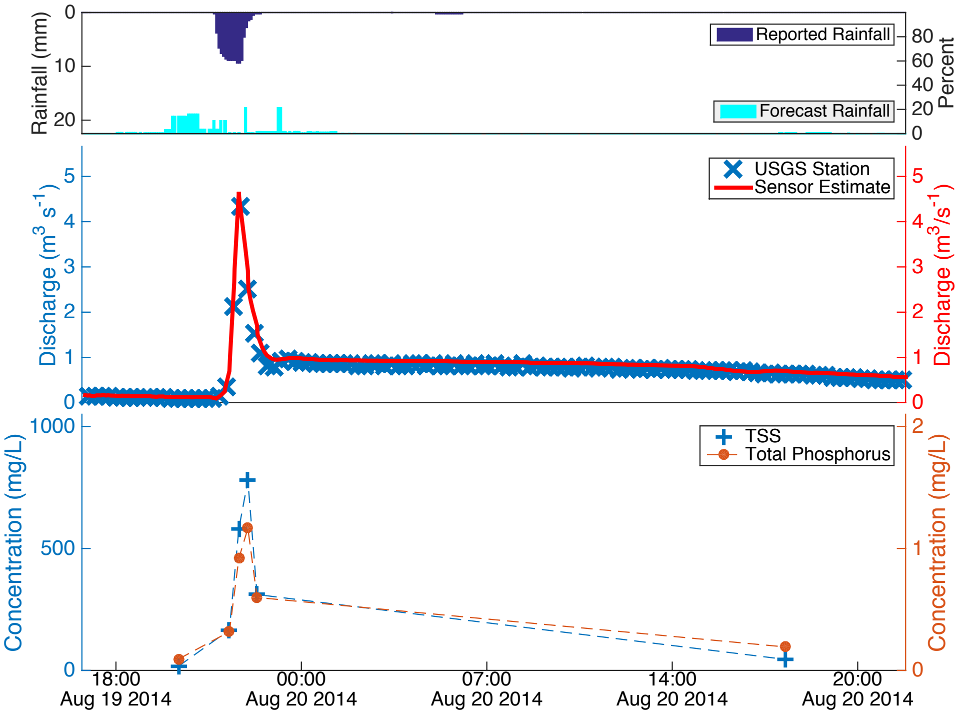
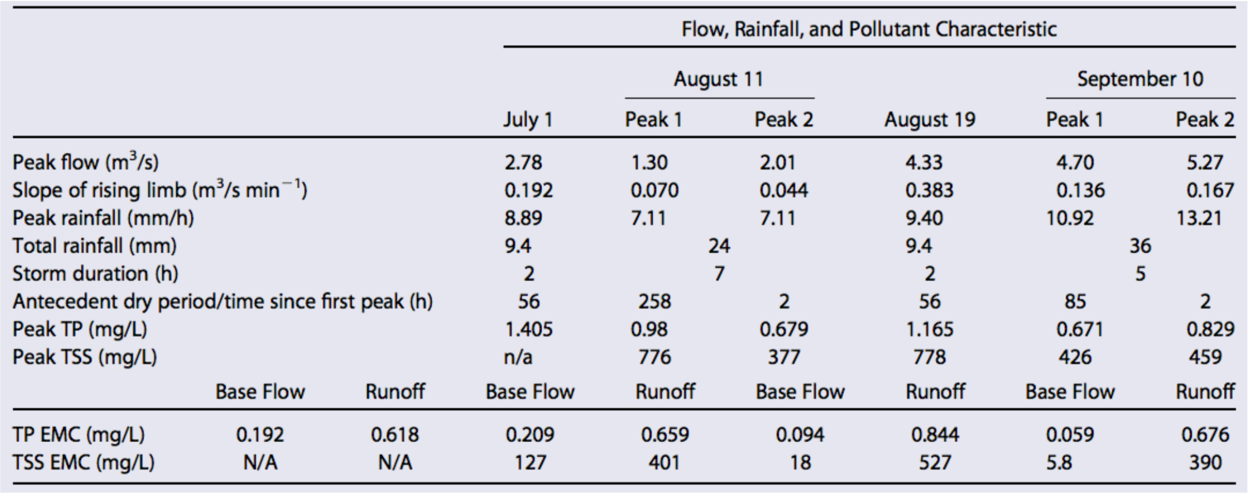


Figure 4.6. Forecasted rainfall and measured hyetograph from *Weather Underground* (top). Hydrograph reported by nearby USGS gage and estimated by local depth sensor (middle). Linearly interpolated pollutograph for total suspended solids (TSS) and total phosphorus (bottom). Markers indicate samples triggered by the algorithm.

4.5.3 Water quality

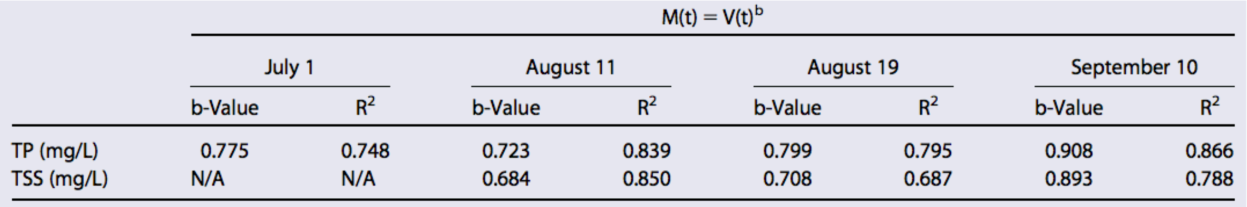
Concentrations for both TSS and total phosphorus showed a positive linear correlation with flow ($R^2 = 0.346$ for TSS; $R^2 = 0.437$ for TP and standard deviations $\sigma = 198.6$ mg/L and $\sigma = 0.272$ mg/L, respectively). Samples taken particularly during peak flows had the highest concentrations and there was no observed hysteresis between peak concentration and peak flow. With the exception of the August 11th event, peak concentrations strictly corresponded with peak flows (Table 4.1). The 7-hour storm event on August 11th drove two distinct discharge peaks. During this event, the largest concentrations occurred during the first peak while the largest flows occurred during the second. In general, for the storms with multiple distinct hydrograph peaks, the intra-storm hydrograph with the relatively steeper rising limb (larger flow acceleration) had the largest pollutant concentration. This was also seen during the September 10th storm event, which also exhibited two distinct discharge peaks. During this event, the second peak, while relatively larger, was also characterized by a steeper rising limb and higher concentrations.



			Flow, Rainfall, and Pollutant Characteristic					
				August 11			September 10	
			July 1	Peak 1	Peak 2	August 19	Peak 1	Peak 2
Peak flow (m^3/s)			2.78	1.30	2.01	4.33	4.70	5.27
Slope of rising limb (m^3/s min^{-1})			0.192	0.070	0.044	0.383	0.136	0.167
Peak rainfall (mm/h)			8.89	7.11	7.11	9.40	10.92	13.21
Total rainfall (mm)			9.4	24		9.4	36	
Storm duration (h)			2	7		2	5	
Antecedent dry period/time since first peak (h)			56	258	2	56	85	2
Peak TP (mg/L)			1.405	0.98	0.679	1.165	0.671	0.829
Peak TSS (mg/L)			n/a	776	377	778	426	459
	Base Flow	Runoff	Base Flow	Runoff	Base Flow	Runoff	Base Flow	Runoff
TP EMC (mg/L)	0.192	0.618	0.209	0.659	0.094	0.844	0.059	0.676
TSS EMC (mg/L)	N/A	N/A	127	401	18	527	5.8	390

Table 4.1. The characteristics for each measured storm event, including peak flow information, rainfall, and the event mean concentrations (EMCs) for total phosphorus (TP) and total suspended solids (TSS).

Temporal comparison of hydrograph and pollutograph peaks showed no discernable leading hysteresis. Similarly, through an M(V) curve analysis, none of the water quality dynamics could be classified as exhibiting a strong *first flush*. Overall, the *b*-values range from 0.684 to 0.908 and $r^2 < 0.9$ (Table 4.2). Six or more samples were collected for each event and M(V) curves were generated using a spline interpolation (Figure 4.7). Similar M(V) curves were observed for both TSS and total phosphorus. TSS could not be analyzed from the July 1 event as the automated sampler was not initially configured to sample a large enough volume to provide aliquots for TSS analysis.



	$M(t) = V(t)^b$							
	July 1		August 11		August 19		September 10	
	b-Value	R^2	b-Value	R^2	b-Value	R^2	b-Value	R^2
TP (mg/L)	0.775	0.748	0.723	0.839	0.799	0.795	0.908	0.866
TSS (mg/L)	N/A	N/A	0.684	0.850	0.708	0.687	0.893	0.788

Table 4.2. The coefficients of determination and b-values for power law functions for total phosphorus (TP) and TSS.

Peak concentrations of TSS and total phosphorus were neither correlated with rainfall intensity ($r^2 = 0.105$ and $r^2 = 0.0277$ for TSS and total phosphorus, respectively) nor antecedent dry weather periods ($r^2 = 0.142$ and $r^2 = 0.0841$ for TP and TSS and total phosphorus, respectively). The largest of the storm events (September 10th, as measured by stage height and cumulative flow volume) recorded the lowest concentrations of TSS and total phosphorus. Overall, the EMC of total phosphorus was at least three times greater during runoff than during baseflow conditions and the EMC of TSS was at least three times greater (Table 4.1). For both TSS and total phosphorus, the runoff EMC of each pollutant did not exhibit a linear trend over time.

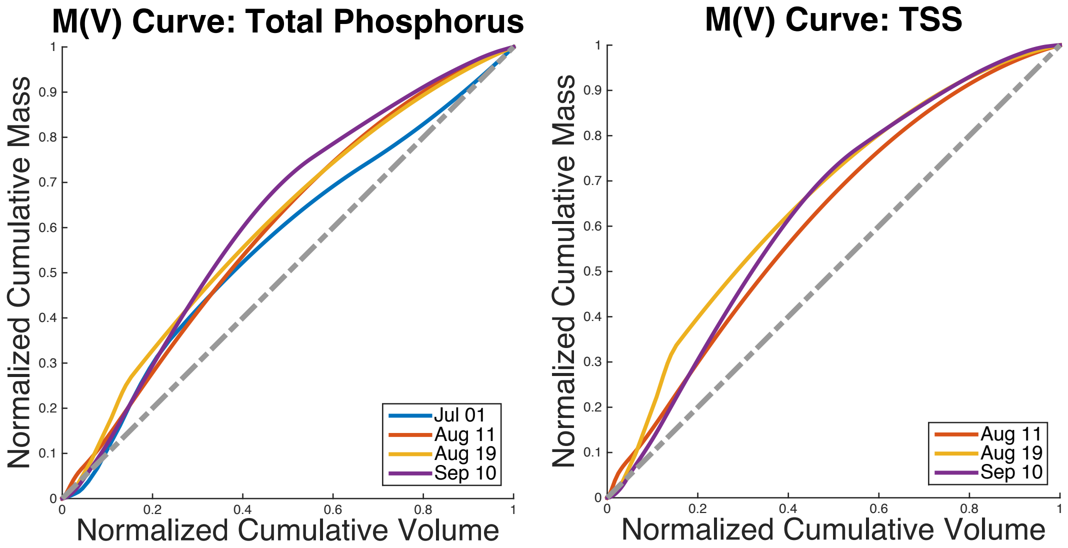


Figure 4.7. Cumulative mass volume curves for total phosphorus (left) and total suspended solids (right). Dashed line indicates uniform pollutant concentration.

4.6 Discussion

4.6.2 Adaptive sampling

Compared to conventional flow- or time-weighted sampling schemes, which rely on preset thresholds, a major benefit of the proposed approach is the ability to anticipate and dynamically adapt to features of an underlying hydrologic signal. This is particularly valuable when resolving pollutograph dynamics across a variety of storm durations and intensities, as it ensures that each distinct hydrograph is characterized using a similar number of samples. Depending on the objectives of the study, this enables the ability to resolve flashy events to the same extent as larger events using the same sampling logic. This not only introduces an element of consistency for inter-storm comparisons, but also reduces the occurrence of missed or excessive samples that are common in conventional sampling approaches. In turn, this improves the use of constrained experimental resources.

If storm patterns drive multiple discharge peaks, such as those experienced on August 11th and September 10th, the smaller peak or the secondary peak, even if short in duration or magnitude, may carry the majority of the pollutant loadings. The use of a flow-weighted approach may have missed such events if parameterized inadequately. A more dynamic estimation approach, as used in this study, is needed to track not only the flow, but also changes in the underlying hydrologic signal. In more advanced experiments, rather than just triggering baseflow samples, the weather forecast could also be used to anticipate the number and timing of samples. In-situ and real-time sensor readings (such as stage or turbidity) will still be required, however, to adapt to site-specific dynamics that cannot be captured by a weather forecast alone. Given the flexibility of our proposed framework, such modifications can be made easily and the sampling logic can be updated in real-time without the need to visit the study site.

The flexibility of the framework proposed in this chapter is perhaps its biggest benefit. While our sampling approach focused on site-specific hydrograph features, the sampling logic could be changed relatively easily to enable a suite of novel and uniquely targeted experimental objectives. Sampling strategies could be modified to detect debris or faulty sensors by tuning the length-scale parameter of the kernel in real-time, or by implementing more complex fault-detection algorithms [152]. Future experiments could also be designed to use distributed rainfall data and measurements from other sensor nodes to optimize sampling around spatial phenomena of interest. For example, sudden changes in flow at upstream sensor nodes could be used to alert downstream nodes or to track a storm as it moves through a region. Additionally, real-time hydrologic models could be used to enable more complex sampling strategies during different seasons. For example, a snowmelt model and a conductivity sensor could be used to guide chlorine sampling during road salting periods. By leveraging an Internet connection, the majority of this control logic could be implemented on off-site computers, improving ease of use by permitting researchers to implement the control logic using systems and languages they are most comfortable with.

The benefits of adaptive water quality sampling can be achieved at a relatively small overhead. In fact, in their simplest implementation, the methods presented in this study could be readily repeated by simply connecting a cellular modem to the autosampler, relying only on a remote computer and public data (for example, streamflow and precipitation obtained from CUAHSI's

Water Data Center[153] to control the sampling schedule. The need to process real-time sensor feeds comes at a slight computational expense, but is well achievable using already existing technologies and data services [154, 155]. By adding in-situ sensors, such as the low-cost water level sensor used in this study, the capabilities of the automated sampler can be extended even further to enable more responsive and complex sampling strategies. For example, given the observed correlation between TSS and total phosphorus, as well as known correlations between those parameters and turbidity [148, 149], an in-situ turbidity sensor could be used to design an adaptive sampling regimen for total phosphorus. Rather than sampling around distinct features of the hydrograph, such a study could focus on sampling around the most uncertain statistical parameters of the regression relationship. This may increase the complexity of the sampling strategy, but it improves the quality of the data input to the regression, and, in turn, the confidence of the statistical relationship.

4.6.3 Water quality

While the occurrence of a first flush may be variable or specifically associated with large and intense storm events [120], no correlation was found between increasing storm intensity and the likelihood of a strong first flush. Similar conclusions have been drawn in other studies that analyzed loading dynamics of urban runoff [104, 126, 129]. The lack of an observable first flush in our watershed could be attributed to a number of causes, including the relatively large size of our study area (28 km^2). Within our study area, a first flush may have existed in much smaller sub-catchments, as suggested by prior studies (less than 1 km^2, see [127]. However, first flush may not be evident for larger watersheds, particularly if the pollutograph travel times for each sub-catchment superimpose, as their confluence may obscure or widen the concentration profile at the outlet of the larger watershed [91, 130]. Furthermore, if one specific area of the watershed contributes the major pollutant runoff, its travel time in relation to peak discharge at the outlet of the watershed could impact the perceived first flush dynamics.

In our study watershed, a large concentration of solids would be expected from the dominant, heavily urbanized and impervious surfaces of the watershed, which all exhibit very short travel times and should have contributed to a first flush if it existed. To that end, it is likely that erosion, caused by flashy hydrographs or high flows, was the primary driver of water quality in the watershed. Studies have shown that the majority of the phosphorus in runoff is sediment-associated [156, 157], but in many highly urbanized watersheds, this may need to be directly confirmed since many management practices are still geared towards treating the first flush [119, 158]. The urban areas in our study watershed may thus not be a major source of nutrient runoff. While outside of the scope of this study, a small number of the events were also analyzed for other dissolved pollutants, which also did not exhibit first flush characteristics.

Although peak pollutant loads corresponded with peak flows, this relationship was nuanced, where a higher fraction of contaminants arrived after peak flow rather than before. This has also been seen in prior studies [104]. Furthermore, b-values were much greater than 0.185, indicating a lack of a strong first flush in our study catchment. As such, flow values may need to exceed geomorphically significant levels to begin moving sediment [11]. However, this would need to be studied in detail by augmenting the sampling strategy.

Peak concentrations were also poorly correlated with rainfall intensity and duration of antecedent dry weather periods. While this is contradictory to some studies [151], it has been observed by others [104]. The relationship between EMC and rainfall has been generally noted to be weak, suggesting that EMC is likely driven by location- rather than storm-specific features [159]. As such, EMC may not be the best sole measure of water quality characteristics, particularly when studying pollutant dynamics of individual storms. Concentrations for any given event are a complex function of buildup and washoff characteristics [104] and spatial rainfall variability, which thus suggests that any given storm event may exhibit unique concentration magnitudes and temporal characteristics.

Throughout this study, pollutograph dynamics were driven by variable storm patterns, a number of which contained multiple hydrograph peaks. Low correlations between concentration and discharge were observed and have been similarly reported for other urban catchments[160], indicating that concentration may not be fully explained by discharge alone. While lower concentrations of TSS and total phosphorus may have resulted from dilution, caused by increased flows mobilizing more coarse-grained sediments [161], this could not be confirmed consistently across all events. Within storm events with multiple peaks, the peak concentration did not just correspond with the peak flow, but rather with the hydrograph peak that had the steeper rising limb (larger flow acceleration). On an intra-event scale, this suggests that rather than a lag in the pollutograph, as would be suggested by M(V) analysis alone, the concentrations are heavily driven by the hydrograph features. The acceleration of flows may correspond with increased forces exerted on solids, which raise the erosive action on the stream. In our watershed, the "*flashiness"* of the hydrograph, a well-known symptom of the *urban stream syndrome* [162], is thus perhaps the best predictor of peak concentrations within an individual event.

Better characterizations of water quality thus demand more spatially dense measurements and an improved understanding of pollutograph dynamics, a task which will be made easier by the adaptive sampling methods presented in this chapter. In particular, more samples will be required to determine if a first flush is evident in smaller upstream locations, where the pollutograph may be dominated by runoff from impervious regions, rather than stream dynamics. That said, up-scaling the adaptive sampling framework will need to be done carefully, as optimal sampling schedules may likely be guided by site-specific features. Even sites that are very close to one another may exhibit distinctly different pollutograph dynamics. As such, initial measurements and calibrations will likely still need to be carried out on each site, after which the most suitable adaptive sampling strategy can be tuned. A feature-driven approach, such as the one presented here, will form a good starting point to help formulate a site-specific sampling strategy. The proposed adaptive sampling framework will provide a flexible and low-overhead means by which to reduce the resources required to investigate the dynamics that are most uncertain at any site.

4.7 Conclusions

Increasing the temporal resolution of measurements will significantly improve our fundamental understanding of water quality dynamics. Understanding these dynamics across various scales can also help decision-making by guiding watershed-specific solutions that strike a balance between local treatment (e.g. green infrastructure), restoration, or end-of-line solutions. Until reliable and cost-effective in-situ sensors are available for most important parameters, multi-bottle automated samplers will continue to provide a strong alternative to resolving the water quality dynamics of hydrologic systems.

Given real-time notifications and the convenience of using a feature-driven approach to automatically collect samples, the method proposed in this chapter could lower barriers for small research groups, agencies or even individuals to now seamlessly maintain large networks of autosamplers (networks of ten or more samplers). The flexibility the framework presented herein not only makes this possible for automated samplers, but also for in-situ sensors that consume a significant amount of power or are limited by reagent availability or electrode duty cycling.

The lack of an observed first flush in our watershed cautions the implementation of many popular stormwater control measures for improving urban runoff quality. The majority of these systems, such as retention ponds and constructed wetlands are designed to capture a maximum volume of flow (one to two year storms), which is retained for settling while excess flows are released through overflow structures. However, if the inflows to the basin do not exhibit a *first flush,* the basin may only retain the initial, lower concentration flows, while discharging higher concentrations once storage capacity has been reached.

An exciting paradigm may arise from this realization however: by equipping urban stormwater systems with sensors and controllers (valves, gates, pumps, etc.), it will be possible to maximize the treatment of runoff through real-time control [163]. While this idea will require significant future studies to vet its promise, the site-specific characterization of water quality dynamics (or corresponding proxies), as provided by our approach, will allow controllers to be optimally tuned to individual storm events. For example, a gate could be opened at the beginning of a storm to allow lower-concentration flows to exit the watershed, while closing to capture the highest concentration inflows and retain them as long as possible before the next storm event. Similarly, these solutions could be implemented upstream to reduce the exceedance of geomorphically significant flows, and thus downstream erosion and nutrient loads. These real-time systems are presently being constructed in this study watershed and will be evaluated in the future.

4.8 Follow-up study: building real-world control networks

Using the lessons learned from controlling an automated sampler (Figure 4.8), a follow up study was carried out to control actual stormwater basins. After some software and firmware modifications, using the same hardware developed for the autosampler, the sensor node was soon transformed into a wireless gateway for controlling valves in real-time. The first basin, located in Ann Arbor, Michigan. was retrofitted with a controllable valve in September 2016 (Figure 4.9) and the first watershed-scale experiment was carried out the following December.

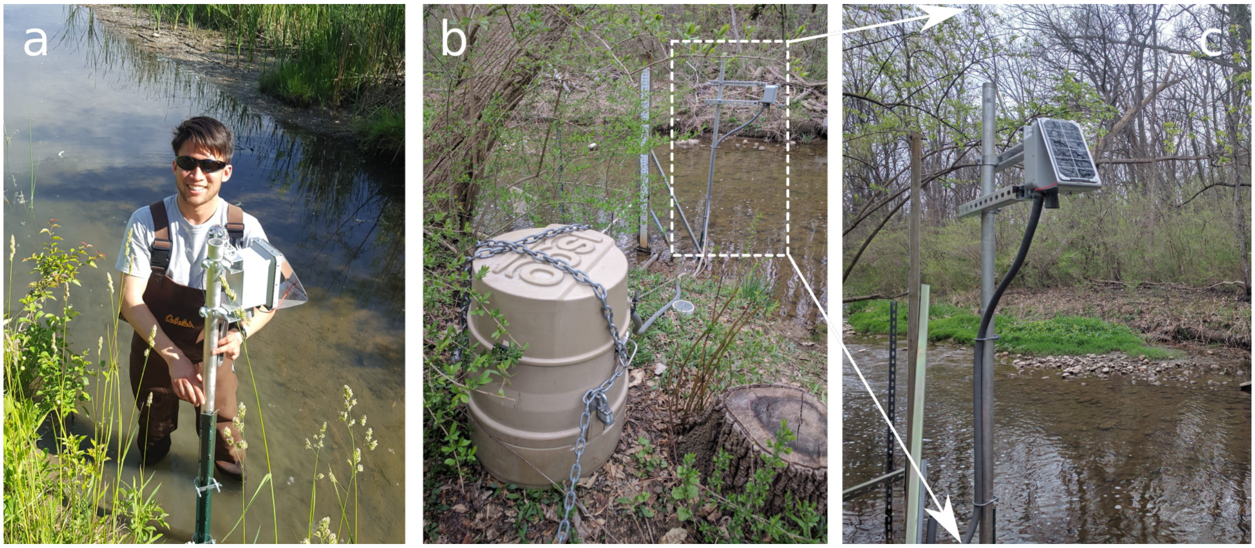


Figure 4.8. Sequence of field deployments for adaptive sampling, with the (a) initial field installation of wireless sensor node, (b) completed assembly with automated sampler, and (c) a close up of the latest design iteration.

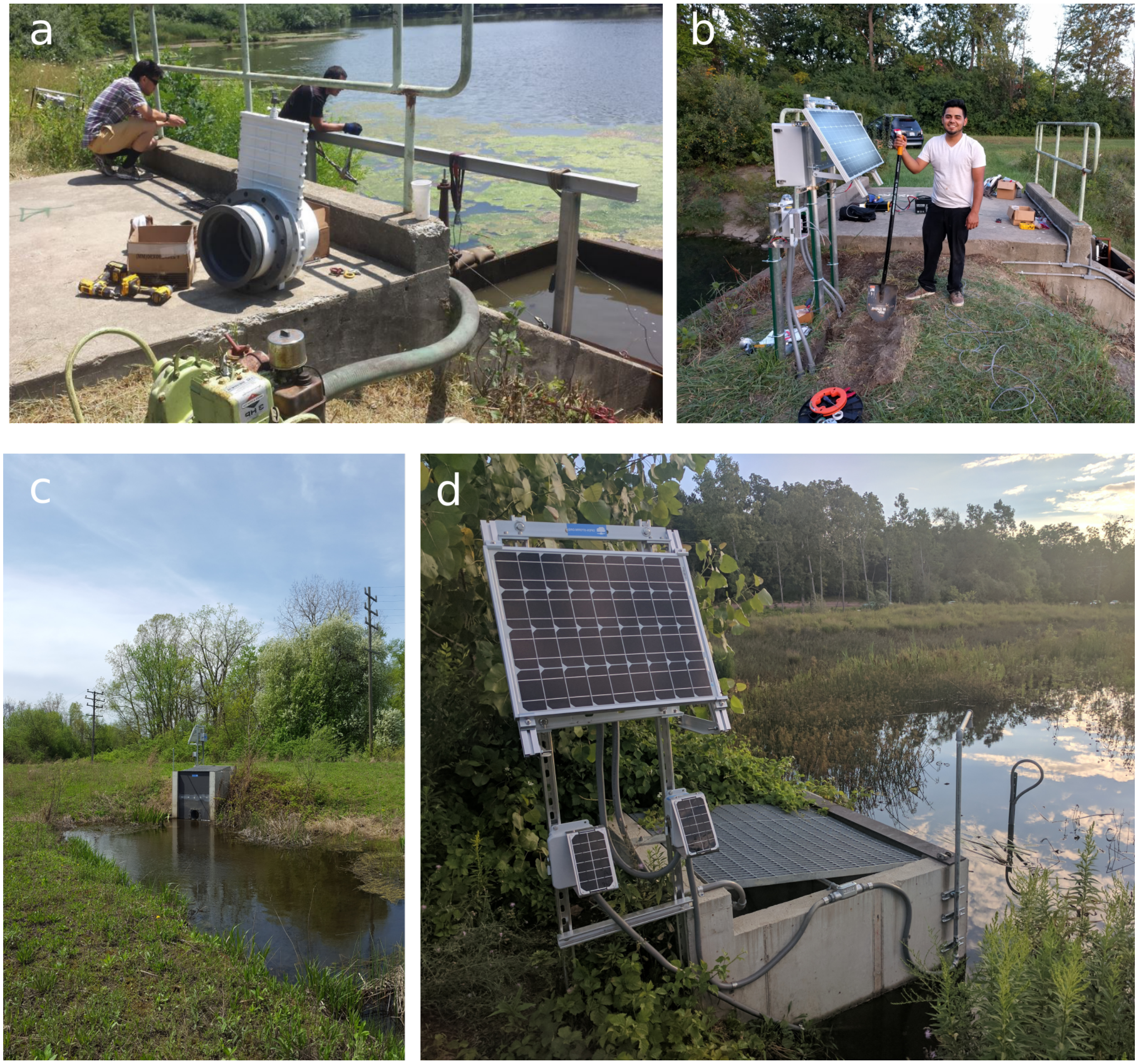


Figure 4.9. (a) Field deployment of a remotely controllable valve at the Ellsworth basin (b) breaking ground installing the electronics, as well as the water level at the County Farm Park basin downstream before (c) and after (d) the installation of another valve. Designs are available freely on Open-Storm.org.

This case study on watershed-scale control was carried out in urban watershed in the City of Ann Arbor, Michigan (Figure 4.5). The Malletts Creek watershed – a 28 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 [164]. 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 was modified for real-time control at two locations by retrofitting existing basin outlets with remotely-operated valves (Figure 4.9). The first control point is a stormwater retention pond in the southern part of the watershed (shown in red in Figure 4.10). 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 4.10). 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. Measurements from the sensor network were validated using an external United States Geological Survey flow measurement station (USGS station 04174518), located at the watershed outlet.

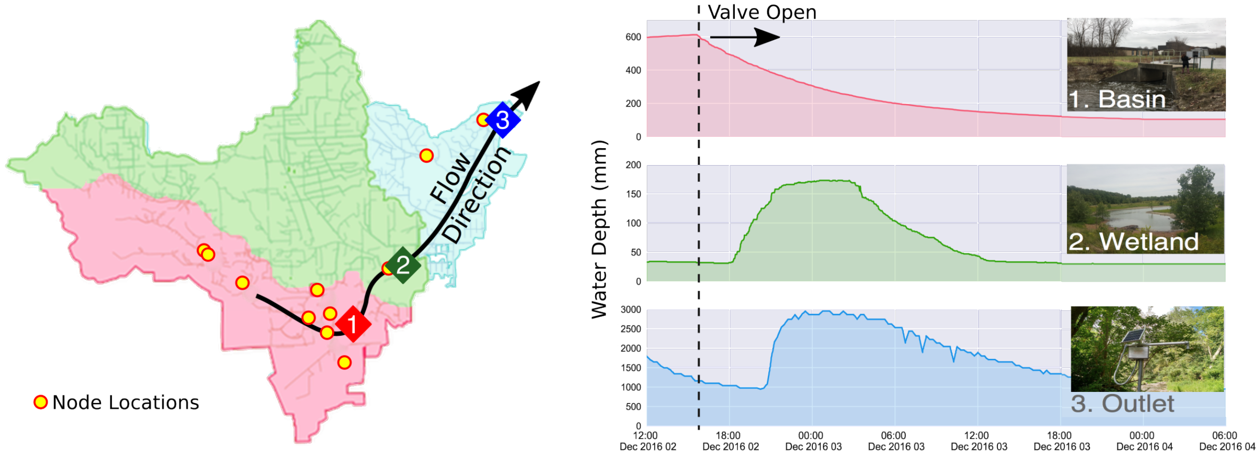


Figure 4.10. Malletts Creek control experiment in Ann Arbor. The left panel shows time series of water depth from 12:00 pm on December 2 to 6:00 am on December 4, 2016. The right panel shows the location of the three sites in the watershed, with the partitioned contributing areas of each location corresponding to the colors of the time series plots.

We confirm the effectiveness of the control network through a simple experiment. In this experiment, stormwater is retained at an upstream control site, and 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 [38] – the goal of this experiment was to test the extent to which control of individual sites can improve watershed-scale outcomes. The 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 4.10 (left) 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 [165], 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 was placed into the wetland to measure the local water level and determine the optimal time to release from the retention pond upstream.

At the onset of the experiment, water was 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 4.10). 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 4.10). 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 [38]. 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 chapters in this dissertation – 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 [165]. 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.

Chapter 5 Real-time control of urban headwater catchments: performance, analysis, and site selection

5.2 Introduction

Having demonstrated the real-world potential of stormwater control, this chapter seeks to determine how control valves can be coordinated across entire watersheds. It is important to note again that population pressures continue to drive land use changes, often resulting in more paved and impermeable urban landscapes. Stormwater runoff has become more flashy and polluted, leading to flooding, erosion, and ecosystem impairments [166]. Often referred to as the *urban stream syndrome* [162], this collection of challenges is compounded by changing climate, which drives storms of increasing intensity and frequency [167, 168]. At a time of declining infrastructure funding [169, 170], pressure is mounting on urban watershed managers to do more with less.

Traditionally, flooding and stream erosion have been mitigated through expansion of constructed stormwater infrastructure, which conveys runoff from buildings and roads through a complex system of below- and above-ground infrastructure, such as pipes, detention basins, and constructed wetlands. Most recently, *green infrastructure*, has risen to prominence in the form of many smaller and distributed assets, such as bioswales, rain gardens, and green roofs [171, 172].

Watershed managers thus have a large portfolio of stormwater options, which from large centralized assets to smaller distributed solutions, most of which are very expensive. Once constructed, stormwater systems are very difficult to adapt to changing land uses and weather. Furthermore, recent studies have shown that aggressive adaptation via many large or distributed stormwater assets can actually lead to worse watershed outcomes if individual elements are not tuned to system-level outcomes [29, 47, 173]. As such, there is an urgent need to find new adaptive solutions that are *aware* of the larger watershed.

“Smart” stormwater systems have recently been proposed to achieve adaptation and system-level control [109, 174]. In lieu of new construction, this paradigm proposes to use many distributed and low-cost sensors and controllers (valves, gates, pumps, etc.) to coordinate flows across the scales of entire watersheds, transforming existing systems to be used much more effectively by adapting them on a storm-by-storm basis. While the technologies to enable this vision have mostly been developed [175], as demonstrated in prior chapters of this dissertation, much fundamental research remains to be conducted to determine how stormwater systems can be controlled safely and reliably across the scale of entire watersheds. This requires an interdisciplinary knowledge of domains spanning hydrology, infrastructure, data sciences, and control theory.

In this chapter we take a step toward the real-time control of urban watersheds by asking the question: Where should urban catchments be retrofitted for real-time control and what performance gains can be achieved compared to *passive* alternatives? The fundamental contributions of this chapter are:

- A feedback control methodology, which mathematically formulates stormwater catchments as dynamical systems and controls them via linear-quadratic (LQ) control.
- A simulation-based approach to help identify how many distributed control valves are needed and where they should be placed to achieve the best real-time control outcomes, focusing specifically on reducing flooding and erosion.
- A holistic equivalence analysis, which compares the real-time controlled system to passive solutions across many storms of varying intensities and durations.

Given that "smart" stormwater systems have yet to be constructed at large scales, this analysis will be carried out in simulation, which will allow for a variety of scenarios to be evaluated before results can be used to build real-world control networks. Furthermore, the analysis will focus on the scale of urban headwater catchments (1-5km^2), which will serve as building blocks to inform the control of larger watersheds in the future.

5.3 Background

Sensors are becoming progressively cheaper [114]. When coupled with now readily-available wireless connectivity, these devices are able stream unprecedented amounts of real-time measurements about the health and performance of large watersheds [165, 175]. Real-time information becomes particularly important when used in a bidirectional fashion. Namely, rather than simply receiving measurements, commands may be transmitted back to watersheds to change flows and hydrologic behavior. A simple example involves the addition of an inexpensive control valve to the outlet of a stormwater basin, such as those currently being retrofitted by the authors in the Midwestern United States [165, 175]. Compared to static solutions, where the outflows are determined by a fixed outlet geometry, real-time control provides the ability to actively modulate runoff and adapt site behavior based on real-time hydrologic states and future forecasts [176].

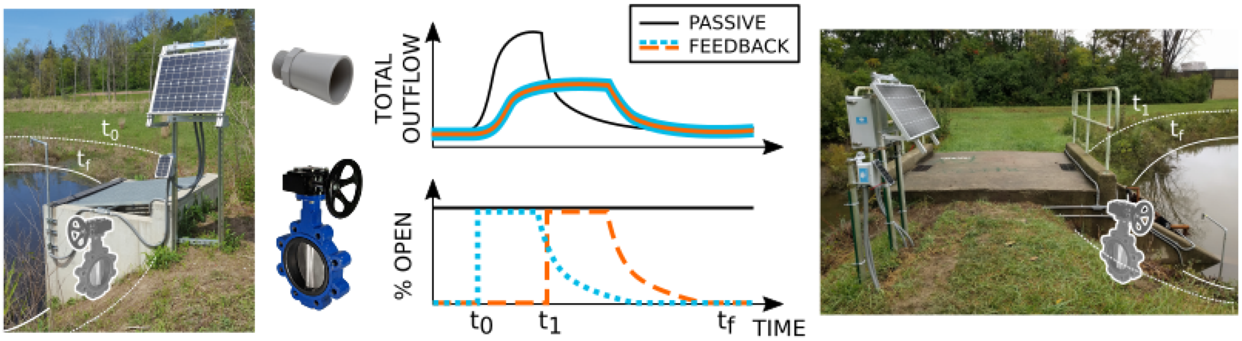


Figure 5.1. Example real-time control sites currently being deployed by the authors. Each site is equipped with water level sensors and a remotely-controllable butterfly valve.

Even just a single remotely-controlled valve can yield significant benefits [42, 177, 178]. For example, a valve can also be used to extend hydraulic retention time, and thus promote the capture of sediment-bound pollutants [41, 44, 179]. Water can then be released if another storm is forecasted or detected to create additional storage capacity. By extension, modulation of flows (hydrograph shaping) from a site could reduce erosion at downstream locations by ensuring that stream flows do not exceed critical downstream levels. Such an approach thus adaptively balances water quality and flooding benefits, which is difficult to accomplish using passive solutions. However, given the recent advent of these technologies, research studies addressing the benefits of real-time control are limited to the site-scale, focusing almost entirely on the control of individual ponds and basins using a single valve.

Perhaps the biggest frontier of real-time stormwater control is the ability to achieve watersheds-scale outcomes. Given the complexity of urban watersheds, which spans an interconnection between hydrology and man-made infrastructure, it presently remains unclear how to orchestrate the operation of multiple controlled sites to achieve watershed-scale outcomes. Guidance on controlling flows across large spatial scales may be taken from seminal research on reservoir operations, open channel irrigation systems, water distribution systems, and sewer systems [49, 175-177]. One of the earliest examples of dynamic control for river systems was the use of linear quadratic control to regulate the daily operation of large hydropower reservoirs [180]. Rather than building control rules into a complex physical model, this study highlighted the benefit of abstracting the physical system into a simpler matrix-based dynamical model, which could be used to apply feedback control. More advanced methods, such as distributed linear quadratic control [181] and model predictive control [182-184], have been successfully applied for the control of canal networks. While these approaches show great promise, they do not explicitly account for the types of time-scales, complexities and feedbacks inherent in urban watersheds. As such, it is presently unclear which real-time control approaches will meet performance goals without risking the safety of nearby residents, property, and downstream ecosystems. This study takes a step toward closing this knowledge gap by formulating and evaluating a dynamical control approach specifically for an urban watershed.

5.4 Methods

5.4.2 Approach

When retrofitting urban watersheds for real-time control, a choice must be made in regard to the spatial scales at which these technologies will first be implemented and analyzed. We contend that the analysis of real-time stormwater control strategies should begin at the scale of urban headwater catchments. These subcatchments are as large as 5 km^2 (2 mi^2), and can be found in most cities, small and large [47, 185, 186]. Overall, the choice to focus on this scale is motivated by a number of fundamental and practical factors. Fundamentally, the scalability of real-time watershed control requires smaller-scale systems to be analyzed and understood first [127]. If feasible at these scales, the control of smaller catchments will ultimately underpin the control of the larger watershed.

Practically, it is unlikely that entire cities will be retrofitted with control valves all at once. Rather, valves will be evaluated one-by-one or as part of controlled clusters. In the United States, decisions to build or upgrade stormwater infrastructure are often driven by new residential or commercial development projects, which impact flows at the scale of local pipe and stream networks [187, 188]. Given the recent emphasis on distributed stormwater management, these measures often include ponds, basins and wetlands at commercial complexes, subdivisions, neighborhoods, and precincts. Urban flash flooding occurs at the scale of local road networks, which suggests that control strategies should operate to prevent flooding even as far upstream as first-order catchments. In fact, the US Federal Emergency Management Agency (FEMA) provides flood advisory and insurance information at scale of 5 km^2 (2 mi^2) sub-watersheds[13], which makes them of particular interest for analysis. Most existing radar and gage rainfall products are offered at 1-5 km^2 resolution as well, which is relevant if rainfall forecasts are to be integrated with real-time control. As such, both fundamental and practical considerations suggest that the scale of headwater catchments (1-5km^2) provides a good starting point to answer the questions posed in this chapter. Future studies can then analyze how the control of larger watersheds can be achieved through lessons learned at the catchment scale.

[13] https://msc.fema.gov/portal/search

5.4.3 Dynamical System Representation

Most modern physical models of urban watersheds, such as EPA's *Stormwater Management Model*[189], are based on a coupled hydrologic-hydraulic approach, where hydrologic dynamics, such as runoff and infiltration, are represented via physical or empirical sub-models. Flows are subsequently routed using a hydraulic engine, typically based on nonlinear *Saint-Venant* equations for shallow water flow [190]. Given the high degree of detail, complexity, and nonlinearities inherent in these models, the application of formal control and optimization approaches becomes intractable. Fortunately, for many complex control systems, such as those used on autopilots and factory processes, perfect models are not necessary to achieve desirable control outcomes. Rather, a control model that approximates the dynamics of the underlying system is often sufficient, since the actual control actions will often steer the system back into domains where the approximations hold true. In feedback control, this is often accomplished by linearizing the system dynamics around desired setpoints (e.g. flows, flood stages, etc.), after which modern control techniques can be applied. For the specific control of water flows in pipes and canals, examples of approximated models have included the integrator delay [191], integrator delay zero models [192], reduced Saint-Venant [193], Muskingum [194], and linear tank models [107].

For our approach, the control model is based on a state-space representation of the hydraulic dynamics as an integrator delay model [191]. In recent studies, this representation has been used for the control of water levels in irrigation canals that are connected in series [193]. However, the use of this formulation for urban watersheds presents additional complexities, making it unclear how well it will work in the control for stormwater systems, if at all. These include the need to accommodate hydrologic effects (runoff, antecedent moisture, etc.) and rainfall, as well as complex and interconnected infrastructure topologies (parallel storage nodes or tree-like networks). Our choice to adopt this approach is based on our expectation that it will sufficiently capture hydrologic and shallow-water flow dynamics. Most importantly, however, the matrix-based representation will allow for the application of modern state-space based control algorithms.

5.4.4 State-space representation of an urban watershed

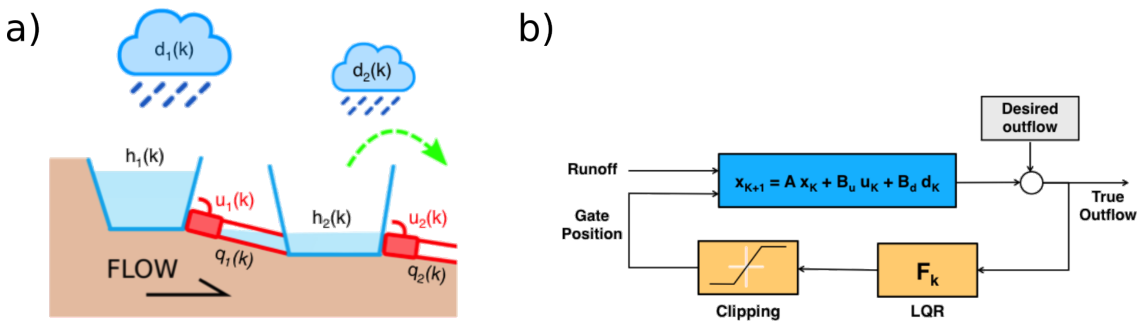


Figure 5.2. Graphical representation of (a) an integrator-delay model and (b) the block-diagram of the feedback controller.

The linearized state-space representation (5.1) of an urban stormwater catchment, modeled as an integrator delay model, can be decomposed into two parts (5.3). The integrator models the change in height of the storage node as a function of the current height h_i(k), inflows q_i(k) from upstream nodes, inflows from local runoff d_i(k), and the controlled outflows u_i(k). The delay component models the travel time of water from one storage node to through a channel to the next node (Figure 5.2a). The full time-varying state-space representation is then given by:

a. The State-space model

$$x(k+1) = \mathbf{A}(k)\, x(k) + \mathbf{B_u}(k)\, u(k) + \mathbf{B_d}(k)\, d(k)$$
$$y(k) = \mathbf{C}\, x(k) \tag{5.1}$$

b. The Integrator component

$x(k+1) = [h_i(k+1)]$, where

$$h_i(k+1) = h_i(k) + \left[\frac{-T}{A_{s,i}(k)}\right] \mathbf{1}_{\{orifices\}}(i)\, Q_{control}(k) + \left[\frac{T}{A_{s,i}(k)}\right] \mathbf{1}_{\{subcatchments\}}(i)\, Q_{runoff}(k) \tag{5.2}$$

$$h_i(k+1) = \boldsymbol{A}_{integrator\ i}(k)\; h_i(k) + \boldsymbol{B}_{u,\,integrator\ i}(k)\, Q_{control}(k) + \boldsymbol{B}_{d,\,integrator\ i}\, Q_{runoff}(k) \tag{5.3}$$

c. The Delay component

$x(k+1) = \begin{bmatrix} Q_i(k+1) \\ Q_i(k) \\ Q_i(k-1) \\ \vdots \\ Q_i(k-n) \end{bmatrix}$, where

$$\begin{bmatrix} Q_i(k+1) \\ Q_i(k) \\ Q_i(k-1) \\ \vdots \\ Q_i(k-n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & & & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Q_i(k) \\ Q_i(k-1) \\ Q_i(k-2) \\ \vdots \\ Q_i(k-n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathbf{1}_{\{orifices\}}(i)\, Q_{control}(k) \tag{5.4}$$

$$\begin{bmatrix} Q_i(k+1) \\ Q_i(k) \\ Q_i(k-1) \\ \vdots \\ Q_i(k-n) \end{bmatrix} = \boldsymbol{A}_{delay\ i}(k) \begin{bmatrix} Q_i(k) \\ Q_i(k-1) \\ Q_i(k-2) \\ \vdots \\ Q_i(k-n-1) \end{bmatrix} + \boldsymbol{B}_{u,\,delay\ i}(k)\, Q_{control}(k) \tag{5.5}$$

d. The Indicator function

$$\mathbf{1}_{\{\xi\}}(a) = [x_1 \quad x_2 \quad \cdots \quad x_n],\ where\ x_i = \begin{cases} 1, & Element\ \xi_i\ flows\ into\ node\ a \\ 0, & otherwise \end{cases} \tag{5.6}$$

The state vector, *x*(k), is composed of the heights, *h*h_i(k), and flows Q_i(k) of all storage nodes in the system at the k^th timestep; the control vector, *u*(k), contains the control outflow $Q_{control,\ i}(k)$

for each controllable orifice/valve; and the disturbance vector, $d(k)$, is a vector of the total runoff $Q_{runoff,\ i}(\mathrm{k})$ from all local subcatchments that flow into the i^{th} storage node. The state matrix, $\mathbf{A}(k)$, relates the heights and flows from the current timestep to the next; the control matrix, $\boldsymbol{B}_u(k)$, links the control outflow to its associated storage node; and the disturbance matrix, $\boldsymbol{B}_d$ routes the rainfall-generated runoff from a given subcatchment to its associated storage node. These matrices are dependent on properties of the physical system, including the sampling period, T, the cross-sectional area of each integrator node at the current timestep, $\boldsymbol{A}_{integrator\ i}(k)$, and the channel characteristics between storage nodes. The output vector, $\mathrm{y}(k)$, is composed of the height and outflow of each storage node while the output matrix $\boldsymbol{C}$ relates the state vector to the output vector. In this study, all states are assumed observable (measured by sensors), allowing us to focus solely on the evolution of the state vector, $x(\mathrm{k})$.

5.4.5 Estimating outflow due to valve opening

To set the outflow for each controlled node $Q_i(\mathrm{k})$ from the control vector $u(\mathrm{k})$, the effective cross-sectional of each valve (how far the valve is opened), $A_{valve}(k)$, was obtained by inverting the equation for flow through a free-flowing undershot gate (5.7) [195].

$$A_{valve}(k) = \frac{Q(k)}{C\mu\sqrt{2gh(k)}} \tag{5.7}$$

Here, $Q(\mathrm{k})$ is the desired controlled outflow; C is a calibration coefficient; μ is the contraction coefficient; g is the gravitational acceleration; and $h(\mathrm{k})$ is the water level of the storage node. To approximate the outflow for an uncontrolled orifice, the equation for flow through a submerged undershot gate was linearized about the height of the storage node (5.8). This was necessary to linearly model the outflow from uncontrolled storage nodes to the connected links each timestep.

$$Q(k) = C\mu A_{valve}(k)\sqrt{2gh(k)} \cong \lambda(k)\, h(k) \tag{5.8}$$

$$\lambda(k) = \frac{gC\mu A_{valve}(k)}{\sqrt{2g\left(h(k) - \mu \cdot \frac{A_{valve}(k)}{Width_{valve}}\right)}} \tag{5.9}$$

5.4.6 Constructing the system matrices from a physical representation

To construct a state-space representation from a physical model or real-world system, the state matrix, $\boldsymbol{A}(k)$, can be assembled by combining the integrator, delay, and link components (5.10):

$$\boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{A}_{integrator\ 1}(k) & \boldsymbol{A}_{link\ 12}(k) & \cdots & & & \\ \boldsymbol{A}_{link\ 21}(k) & \ddots & & & & \\ \vdots & & \boldsymbol{A}_{integrator\ m}(k) & & & \\ & & & \boldsymbol{A}_{delay\ 1}(k) & & \\ & & & & \ddots & \\ & & & & & \boldsymbol{A}_{delay\ n}(k) \end{bmatrix} \tag{5.10}$$

where the link component is derived from the linearized equation for flow through a submerged undershot gate (5.9). The link component joins the integrator and delay components (5.11):

$$\boldsymbol{A}_{link\ i,j}(k) = \begin{bmatrix} 0 & \cdots & a(k) \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}, \text{where } a(k) = \begin{cases} 1, & \boldsymbol{A}_i,\ \boldsymbol{A}_j \text{ are delay terms} \\ \frac{T}{A_{s,i}(k)}, & \boldsymbol{A}_i \text{ is an integrator term} \\ 0, & \text{otherwise} \end{cases} \tag{5.11}$$

The control matrix, $\boldsymbol{B}_u(k)$, is similarly assembled by iterating through all storage nodes and vertically concatenating the integrator components, $\boldsymbol{B}_{u,\,integrator\ i}(k)$, followed by concatenating the delay components, $\boldsymbol{B}_{delay}(k)$. The disturbance matrix, $\boldsymbol{B}_d$, is assembled by iterating through all storage nodes and vertically concatenating the integrator components, $\boldsymbol{B}_{u,\,integrator\ i}(k)$, and padding the remaining rows with zero for the delay components:

$$\boldsymbol{B}_u(k) = \begin{bmatrix} \boldsymbol{B}_{u,\,integrator\ 1}(k) \\ \vdots \\ \boldsymbol{B}_{u,\,integrator\ m}(k) \\ \boldsymbol{B}_{u,\,delay\ i}(k) \\ \vdots \\ \boldsymbol{B}_{u,\,delay\ n}(k) \end{bmatrix} \quad (5.12) \qquad \boldsymbol{B}_d(k) = \begin{bmatrix} \boldsymbol{B}_{d,\,integrator\ 1}(k) \\ \vdots \\ \boldsymbol{B}_{d,\,integrator\ m}(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.13)$$

5.4.7 Control algorithm

A linear-quadratic regulator (LQR) was used to control the outflows from each controllable storage node (Figure 5.2b). Linear-quadratic (LQ) control is a matrix-based, closed-loop feedback control method that incorporates open loop dynamics to achieve desired set points [196]. In this case, set points include water level and flows throughout the system. LQR is suitable for real-time control since the matrix computations are relatively fast, making it possible to run even on modern microcontrollers. LQR controls a system described by linear differential equations (5.1) with respect to a quadratic cost function *J*:

$$J = x^T(N)\,\mathbf{Q}\,x(N) + \sum_{k=0}^{N-1} (\rho \cdot x^T(k)\,\mathbf{Q}\,x(k) + u^T(k)\,\mathbf{R}\,u(k)) \qquad (5.14)$$

Cost matrices, **Q** and **R**, need to be tuned to generate a cost function that produces results aligned with the desired setpoints. The parameter *ρ* shifts the weight of the cost between states and control inputs. Over the period which the system dynamics are constant, the performance cost of LQ control for linear, time-invariant systems, if it exists, is optimal, minimal and bounded [197, 198]

Given the cost matrices and assuming the state and input matrices $(\mathbf{A}(k),\ \mathbf{B}(k))$ from the state-space representation are stabilizable, the controlled outflow for each storage node is then given by the control vector:

$$u(k) = -\mathbf{K}\mathrm{x}(\mathrm{k}) \tag{5.15}$$

where **K** is the gain matrix:

$$\mathbf{K} = (\mathbf{B}_{\mathbf{u}}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{B}_{\mathbf{u}}(k) + \mathbf{R})^{-1}(\mathbf{B}_{\mathbf{u}}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{A}) \tag{5.16}$$

and the *cost-to-go* matrix **P** is the solution to the discrete time *Ricatti* equation:

$$\mathbf{A}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{A}(k) - \mathbf{P} - \left(\mathbf{A}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{B}_{\mathbf{U}}^{\mathbf{T}}(k)\right)(\mathbf{B}_{\mathbf{u}}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{B}_{\mathbf{u}}(k) + \mathbf{R})^{-1}(\mathbf{B}_{\mathbf{u}}^{\mathbf{T}}(k)\,\mathbf{P}\,\mathbf{A}) + \mathbf{Q} = 0 \tag{5.17}$$

To ensure a gain matrix could be computed for every valve placement combination, the system is factored into controllable and uncontrollable components using Kalman decomposition [199]. The transformed system is obtained by applying a linear transformation **T**, where:

$$\hat{x}\,(k) = \begin{bmatrix} \hat{x}_C\,(k) \\ \hat{x}_{NC}\,(k) \end{bmatrix} = \mathbf{T}x(k) \tag{5.18}$$

$$\hat{\mathbf{A}}\,(k) = \begin{bmatrix} \hat{\mathbf{A}}_C\,(k) & \hat{\mathbf{A}}_{12}\,(k) \\ \hat{\mathbf{A}}_{21}\,(k) & \hat{\mathbf{A}}_{22}\,(k) \end{bmatrix} = \mathbf{TA}(k)\mathbf{T}^{-1} \tag{5.19}$$

$$\hat{\mathbf{B}}_{\mathbf{u}}\,(k) = \begin{bmatrix} \hat{\mathbf{B}}_{\mathrm{u},C}\,(k) \\ \mathbf{0} \end{bmatrix} = \mathbf{TB}_{\mathbf{u}}(k) \tag{5.20}$$

where matrices $\widehat{\boldsymbol{A}}_{\boldsymbol{C}}(k)$and $\widehat{\boldsymbol{B}}_{u,C}(k)$ are the controllable components. Applying the same transformation to the cost matrices yields:

$$\hat{\mathbf{Q}} = \mathbf{TQT}^{-1} \tag{5.21}$$

$$\hat{\mathbf{R}} = \mathbf{R} \tag{5.22}$$

Since the resulting pair ($\widehat{\boldsymbol{A}}_C(k), \widehat{\boldsymbol{B}}_{u,C}(k)$) is stabilizable, the gain matrix $\widehat{\boldsymbol{K}}$ can be obtained for the controllable subsystem using transformed cost matrices and the same method (5.16). Finally, the control input can be computed as:

$$u(k) = \begin{bmatrix} \hat{u}\,(k) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{K}}\,\hat{x}\,(k) \\ \mathbf{0} \end{bmatrix} \tag{5.23}$$

The form of the resulting control signal *u*(k) remains unchanged and contains the outflow of each controllable valve just as before.

5.4.8 Enforcing physical constraints

Before applying the controller inputs, a clipping component was used to enforce real-world constraints on the outflow and opening of each valve (Figure 5.2b). At each timestep, once all control outflows were calculated, values were constrained to a nonnegative range limited by the maximum allowable outflow. The clipping function for flow is given by a piecewise function:

$$u_i(k) = \begin{cases} Q_{i,max}, & u_i(k) > Q_{i,max} \\ u_i(k), & 0 \leq u_i(k) \leq Q_{i,max} \\ 0 & u_i(k) < 0 \end{cases} \tag{5.24}$$

Once the control outflows were clipped, they were transformed using (5.24) to determine the opening of each orifice. A second clipping component was used to constrain each orifice to a nonnegative area limited by the maximum orifice area. Once these values were determined, they were used to set the state of each valve.

5.5 Implementation

5.5.2 Physical modeling

Many studies often evaluate the performance of control algorithms on the linear models they are based on. If this simplified linear model does not adequately capture the physical hydraulic-hydrologic dynamics, however, it may give the impression that the controller performs better than it actually would in the real-world. To address this concern, our approach applied the linear controller to a physical model. In this fashion, the linearized model is used to make control decisions, while the physical model reflects what real-world and nonlinear outcomes may be. Control performance was evaluated using the US Environmental Protection Agency Stormwater Management Model (SWMM), a popular hydrologic-hydraulic computational model that has been successfully used in the planning, analysis and design of urban drainage systems [189]. SWMM numerically solves the one-dimensional Saint-Venant equations to accurately model transient surface runoff and open-channel flow. Stormwater *hydrology* is modeled based upon a collection of homogeneous sub-catchment areas that receive precipitation and generate runoff, while stormwater *hydraulics* are modeled by routing runoff through a network of channels, storage units, and orifices. Although SWMM is computationally more complex than our integrator-delay model, accurately modeling the water levels in the storage nodes and channel flows is vital to understanding the proposed algorithms may actually perform when subjected to real-world physics.

While the SWMM model provides a powerful simulation engine and rudimentary control rules (e.g. site-scale water level control), it was not designed to be use with system-level control algorithms, such as the one in this study. To that end, we implemented a customized modeling framework that uses the SWMM engine, but executes the model in a stepwise fashion [200]. Rather than running the model for the duration of an entire storm, the model is halted every time step, after which the states can be extracted and an external logic module can be used to set the states of valves and gates across the entire system. Since the physics engine, which is written in the *C* programming language, is implemented as a stand-alone library, the framework provides a wrapper to interface SWMM with modern and popular languages, including Python and Matlab. This allows for the seamless interaction of modern computational and control libraries with the physical modeling of SWMM without necessarily having to implement the controller in the original SWMM model itself.

The first step in the simulation process involves the abstraction of the physical watershed into a linearized control model. While this can be achieved manually, on a case-by-case basis, our approach automates this by first extracting physical parameters from a SWMM model and then converting them to the state-space formulation. Constructing a state-space representation of a SWMM model begins with importing the properties of the SWMM model, including storage curves, contributing subcatchments, and the connectivity between links (pipes, channels, etc.) and storage nodes. These properties are then used to build the state, control, and disturbance matrices $\mathbf{A}(k)$, $\mathbf{B}_{\mathrm{u}}(k)$, and $\mathbf{B_d}$. The construction of these matrices is detailed in Algorithm 5.1 and Algorithm 5.2.

```
storages, junctions, conduits ← load_swmm_model()
T  ← simulation timestep
$A_s$ ← average pond area
n  ← 1

for each node in storages
    A(n,n) = A_integrator_matrix(T, $A_s$)
    n ← n+1
endfor

for each node in junctions
    A(n,n) = A_delay_matrix(T, $A_s$)
    n ← n+1
endfor

for each link in conduits
    m,n ← indices_of( link.head_node, link.tail_node )
    if head_node(link) is in storages
        A(m,n) = A_link_matrix( lambda(k) )
    else
    if head_node(link) is in junctions
        A(m,n) = A_link_matrix(1)
    endif
endfor
```

Algorithm 5.1. Algorithm to construct the state matrix, **A**, using properties from a SWMM model.

```
storages, junctions, conduits ← load_swmm_model()
n ← 1

for each node in storages
    Bu(n) = integrator_control_matrix(T, $A_s$)
    Bd(n) = integrator_disturbance_matrix(T, $A_s$)
    n ← n+1
endfor

for each node in junctions
    Bu(n) = delay_control_matrix(1)
    Bd(n) = 0 * delay_control_matrix(1)
    n ← n+1
endfor
```

Algorithm 5.2. Algorithm to construct the control matrix, $\mathbf{B}_u$, and disturbance matrix, $\mathbf{B}_d$.

If the physical model is not controlled, it can be executed in a stepwise fashion in Matlab or Python using Algorithm 5.3. The states of the model (water levels, flows, etc.) can then be extracted or visualized and the model can be halted once a specific state has been reached or total duration has been exceeded. For the controlled case, the model is halted every step, after which the control matrixes of the linear model are updated (Algorithm 5.4). The outflow for each storage node is given by the control gain computed via an LQR control function and clipped, if necessary. The valve and gate positions are then set as a relative percentage of the total area. The loop is then repeated until the simulation period expires or specific state is reached.

```
while simulation is not over
    swmm.step_forward
    k ← k+1
endwhile
```

Algorithm 5.3. Executing a SWMM model simulation without a controller in Matlab using the MatSWMM toolbox.

Algorithm 5.4. Implementing the LQR control algorithm in Matlab using the MatSWMM toolbox.

```
get_properties(swmm)

while simulation is not over
        swmm.step_forward
        xhat[k] ← T * swmm.get_states()
        Ahat, Bhat, Qhat, Rhat ← update_system_matrices(xhat[k])
        Khat ← dlqr(Ahat, Bhat, Qhat, Rhat )
        Qoutflow ← Khat * xhat[k]
        Qclipped ← clip( Qoutflow, 0, Qmax )
        PercentOpen ← clip( Agate(Qclipped)/ Amax, 0, 1) * 100
        swmm.update_gate_positions(PercentOpen)
        k ← k+1
endwhile
```

5.5.3 Study Area

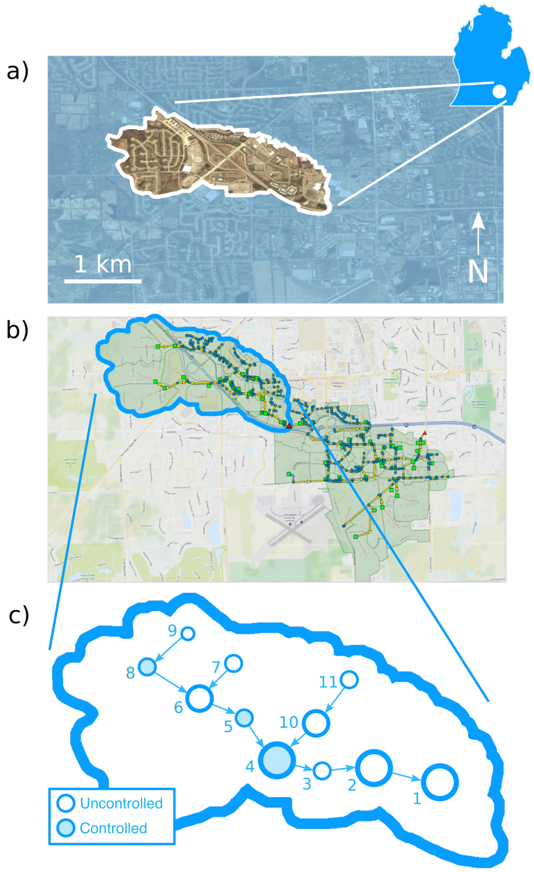


Figure 5.3. The (a) study catchment, (b) model representation in SWMM, and (c) network representation for the linearized model.

The control approach was simulated on a 3 km^2 catchment in Southeast Michigan, which is presently under consideration for real-time control (Figure 5.3). This catchment has been of particular interest to local officials due to stream erosion, who have been calling for improved means to reduced flows at the outlet of the watershed. The catchment is comprised of 11 basins or ponds, ranging in volume from 370 m^3 to 32000 m^3. A calibrated SWMM model of the catchment was made available to the authors by city managers, thus reflecting the most up-to-date knowledge of the real system. To represent valves, each storage node in the model was retrofitted with an adjustable 0.1 m^2 (12 inch) orifice, located at the outlet of the site. Each orifice had a higher invert elevation than its overflow height and all conduits between storage nodes were circular in geometry, with a maximum diameter of 0.3 m. A Green-Ampt model was used to model soil infiltration in the subcatchments [201].

A linear control model was formulated from the SWMM model using the methods described previously. For the catchment in this study, $\mathbf{A}(k)$ was a 55 x 55 matrix; $\mathbf{B}_{\mathrm{u}}(k)$ was a 55 x 11 matrix; and $\mathbf{B}_{\mathrm{d}}$ was a 55 x 19 matrix. The state vector $\mathrm{x}(k)$ was a 55 x 1 vector and the control vector *u(*k*)* was an 11 x 1 vector. The transformation matrix, $\mathbf{T}$, was obtained using the *minreal*() function in Matlab. Finally, applying the transformation to decompose the integrator-delay model and isolate the controllable components, the *dlqr*() function was used in *Matlab* to compute the gain matrix, $\widehat{\boldsymbol{K}}(k)$ =*dlqr(*$\widehat{\boldsymbol{A}}_{\boldsymbol{C}}(k), \widehat{\boldsymbol{B}}_{\boldsymbol{u},\boldsymbol{C}}(k), \widehat{\boldsymbol{Q}}, \widehat{\boldsymbol{R}}$). The resulting linear representation for this catchment has been attached in the appendix. While the model executed at a five second resolution, control actions were constrained to five minute windows to be consistent with the sensor and control networks currently being deployed by the authors [202]. All simulations were carried out on a high-performance Linux cluster at the University of Michigan.

Following common practice in stormwater engineering, the modeled catchment was subjected to a variety of synthetic *design storms* [203]. To account for storms of various intensity and duration, the physical model was simulated with rainfall from *Soil Conservation Survey* (SCS) Type-II design curves [204], which are commonly used in the United States infrastructure design [205]. Statistical storm data provided by NOAA Atlas 14 was used to define the intensity of the storms for a given storm duration and return period [206]. For example, a design storm of 24-hour duration with a 10-year return period, henceforth referred to as a 10-year, 24-hour storm, has an average cumulative rainfall of 83 mm. The storms were sampled at five-minute resolution and used inputs into the simulation.

5.6 Performance evaluation

To evaluate the performance of the LQR-based feedback controller, a baseline performance objective was first established by evaluating how the uncontrolled system responds to a relatively small event (2-year, 24-hour storm). During this storm, there were no overflows at any of the storage nodes and the peak flows in the catchment reached 0.3m^3/s at the outlet. It was then evaluated if the controlled system could reach the same baseline performance during larger events. This was intended to reflect the benefits of control in terms of flooding, as well as stream erosion, which is often triggered through the exceedance of geomorphically significant flow magnitude [207-209]. This reasoning also aligns with many current infrastructure design philosophies, which seek to capture larger storms and release them as smaller storms [210-212] (e.g. capture a 10-year event and release with outflows comparable to a 2-year event prior to the addition of control measures). To maintain a clear relationship between tuning parameters and LQR control performance, the tuning parameter ρ was set to 3500 following manual tuning during the 10-year storm, while the **Q** and **R** matrices were set to be identity matrices. In this configuration, **Q** was a 55 x 55 identity matrix, and **R** was an 11 x 11 identity matrix.

The performance was evaluated across the entire system by combining the volume of flooding along with the flow exceedance ($Q^*_{out} = 0.29$ m^3/s) across the duration of an entire storm. Specifically

,

$$P = \sum_{nodes} \sum_{timesteps} Q_{i,flood}(k) + \alpha \cdot max(Q_{i,outflow}(k) - Q_{out}^{*}, 0) \tag{5.25}$$

The weighing α can be tuned to reflect the relative importance of each objective (flooding vs. erosion, for example). In this analysis, α was chosen to be 0.1 to scale the outflows to have the same magnitude as the overflows. First, the performances of the controlled and uncontrolled systems were compared for a 10-year, 24-hour event, assuming that all eleven storage nodes were controlled. In our study area, this event is designated by regulatory guidelines as the design storm that all new developments must meet [213]. Next, using the same storm, all 2048 possibilities of controlled configurations were evaluated, ranging from only one site being controlled to all eleven sites being controlled. These configurations were then ranked to determine which specific configuration provided the best performance, seeking to identify which sites and features may be indicative of good performance. This search was then repeated for 5-, 25-, 50-, 100-, and 200-year, 24-hour storms to confirm if the same configuration was consistent for larger storm events of similar duration. Once the top configuration was identified, its performance was compared to the uncontrolled case over a comprehensive array of storms, raging across 5-, 10-, 25-, 50-, 100-, and 200-year return periods, across durations from 15 minutes to 24 hours. Finally, to investigate how much smaller storage volumes could be constructed when control is used, the volume of the controlled storage nodes was reduced until the overall catchment matched the performance of the uncontrolled systems.

5.7 Results

5.7.2 Performance for the fully-controlled network

On average, when all storage nodes (SNs) were controlled, the LQR-based control approach outperformed the uncontrolled system, both at the scale of individual sites as well as at the watershed outlet. Specifically, when evaluated on the 10-year 24-hour all but one of the SNs did not overflow, nor did their individual outflows exceed the critical flow level of 0.3 m^3/s. Plotted in (Figure 5.4) is a dynamic comparison for controlled and uncontrolled SNs across the system for a 10-year storm. The outflows of the uncontrolled SNs exhibited the familiar hydrograph shape, with a distinct peak and recession period. The outflow hydrographs from the controlled sites exhibited a table-like shape, where flows were controlled up to the desired setpoint and maintained for the duration of the storm event. This also resulted in longer retention times and higher water levels in the controlled system since water was held in SNs so as to not exceed the outflow threshold.

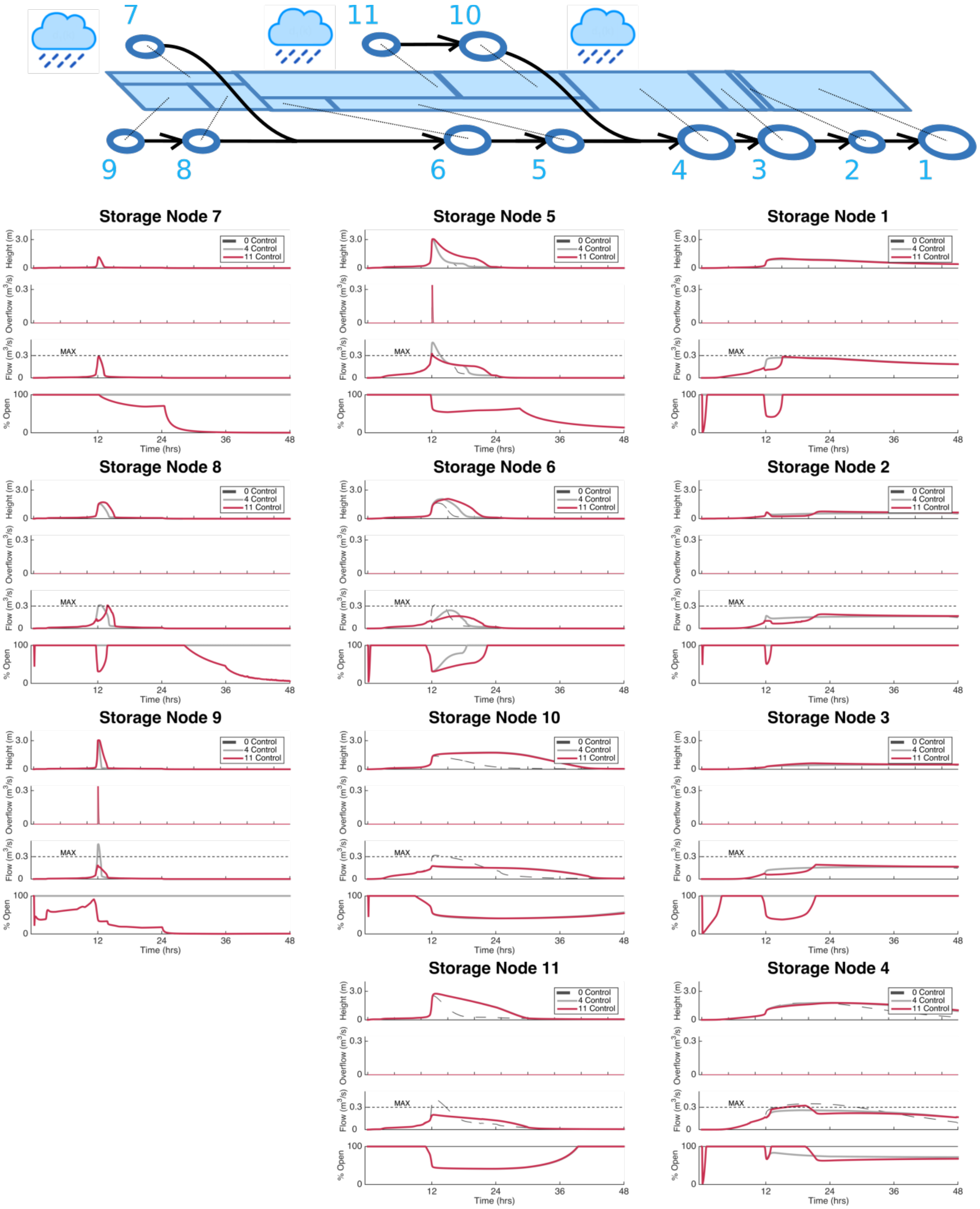


Figure 5.4. System diagram of storage nodes and their response in height, flow, overflow, and valve opening. Labeled are cases with zero controllers (no control, dashed line), four controllers, and eleven controllers. In the case with four controllers, nodes 4, 6, 10, and 11 were controlled. The connected box represents the fractional area of the total subcatchment where rainfall flows to the given node.

5.7.3 Best individual control sites

For the 10-year, 24-hour storm, when only one storage node (SN) was controlled at a time, six of the eleven possible sites showed a notable improvement compared to the uncontrolled system, as measured by the performance across the entire catchment (5.25). The control of one site, in particular (SN4) exhibited a significant relative improvement. Control of three of the eleven SNs did not result in improvements compared to the uncontrolled system because the uncontrolled system already met the control objectives. Only one site (SN9) performed worse when controlled, exhibiting local flooding compared to the uncontrolled case. This SN had the smallest storage capacity in the entire systems, but a relatively large contributing catchment area. Adding a controller led to closure of the valve at the onset of a storm, after which the storage node filled up and could not be drained fast enough once the peak of the storm arrived.

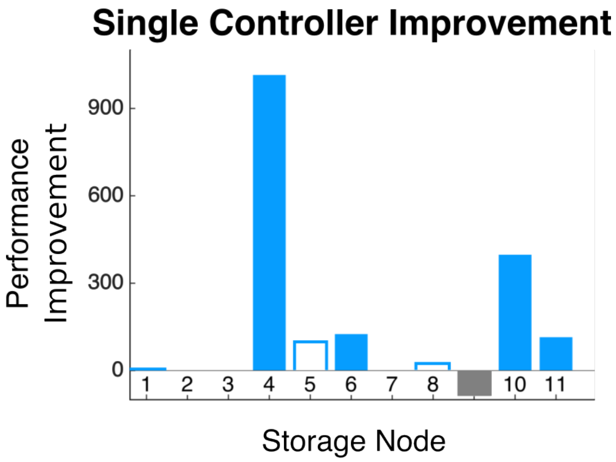


Figure 5.5. Performance improvement, as evaluated by equation (5.25), compared to baseline when only one valve is controlled at a time.

5.7.4 Impact of increasing the number of controlled sites

The addition of the first control valve added the biggest benefit. The addition of more control valves improved the performance, but each successive valve led to marginal returns (Figure 5.6).

The major exception to this trend occurred when the control network was expanded to include ten and eleven controlled SNs, which resulted to slight degradation in performance due to local flooding at smaller storage nodes. In all, over 15,000 simulations (carried out across all possible 2048 possible control configurations and over 24-hour 1-, 2-, 5-, 10-, 25-, 50-, 100-, 200-year storms) showed that adding more control valves improved the relative performance of the overall system, as measured by the cost function.

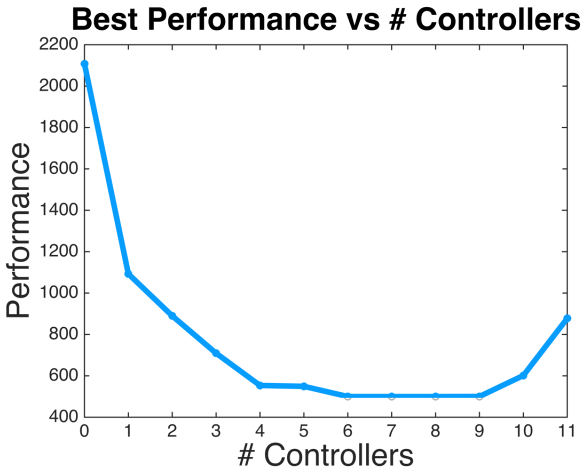


Figure 5.6. The best performance achieved with a given number of controllers for a 10-year, 24-hour storm event.

When analyzing the performance of all possible 2048 control configurations, SNs 4, 6, 10, and 11 consistently appeared in configurations that ranked in the top ten percent (Figure 5.7). Interestingly, these same SNs were those showed the relatively best performance when controlled individually. Overall, out of all 2048 possible combinations of controlled sites, the control of SNs 4, 6, 10, and 11 resulted in the best performance for the 10-year, 24-hour storm using the least number of control points. No physical features (volume, location in watershed, etc.) were consistent across SNs 4, 6, 10, and 11 as an explanation as to why these sites performed the best. The performances of this particular control configuration are compared to the uncontrolled system in Figure 5.8.

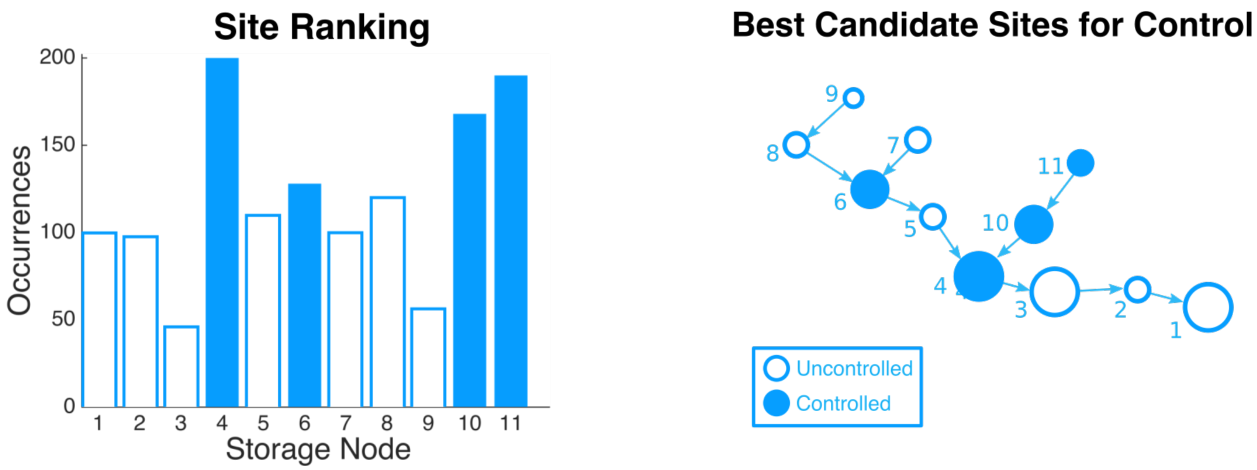


Figure 5.7. The ranking and location of the sites that appeared most in the top ten percent of performances for all controller combinations.

A comparison of the same control configuration (SNs 4, 6, 10, and 11) to the uncontrolled system across a variety of storms (5-200 year, and 15 min, 24-hour storms), showed a better performance. Overall, the number of cases, or zones, associated with relatively high performance (no overflows and low outflows) expanded notably for the controlled case, as indicated by the larger dark blue region in Figure 5.8. In fact, during approximately a 30-year, 24-hour storm, the controlled system had the same performance as the uncontrolled system during a 10-year, 24-hour event. In other words, the controlled system was able to handle much bigger storms, without compromising performance. Furthermore, it was determined that the SNs that were controlled could be reduced in volume by over 50% and still achieve the same performance as the original uncontrolled system.

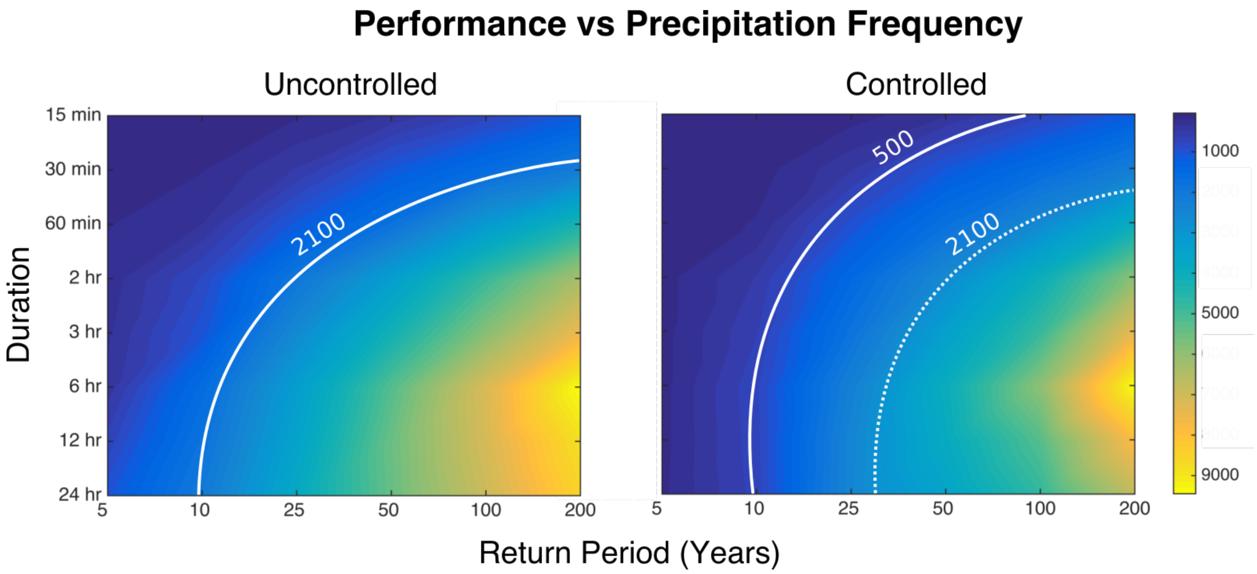


Figure 5.8. System performance for a range of design storms with various intensities. Nodes 4, 6, 10, and 11 are controlled in the controlled case. The controlled system "shifts" the performance zone notably, achieving the same performance for a 30-year storms as the uncontrolled system for a 10-year storm.

5.8 Discussion

5.8.2 Control performance

Despite the hydraulic and hydrologic complexity inherent in urban stormwater catchments, a linear-quadratic feedback controller was able to significantly improve the modeled flows in this study. It is important to note that there was, at the onset, no guarantee that this was going to be the case, since the linear representation of the control dynamics may have seemed over-simplified compared to more traditional and physically-based stormwater models. This validates the use of feedback-based methods, even for the control of complex and nonlinear stormwater catchments. It is likely that nonlinearities may become more important as the scale of the controlled systems grows. While this may be worth exploring in future studies, it may result in the need for more complex control approaches. Rather, the authors contend that the control of larger watersheds could be achieved by controlling individual sub-watersheds and setting their outflows to meet a cumulative goal at the larger scale. As alluded to earlier, this may also be more realistic, since many management and design decisions occur at these smaller headwater scales. To that end, this approach shows great promise at the urban headwater scale (1-5km^2). While the deployment of real-time controllable stormwater valves is still growing, the results of our simulations suggest that the control of entire urban watersheds may be very feasible future goal.

In our study, the control objectives were tuned to reducing outflows and flooding, but the controller and cost function could readily be extended to meet other goals. For example, systems could be tuned to maximally retain water by keeping storage levels near capacity. This would increase hydraulic residence time after storms and thus help with the treatment of sediment-bound and dissolved pollutants. Our study also showed how outflows from the catchment could be "shaped" beyond a traditional hydrograph. In this study, the controlled hydrograph was flat for the majority of its duration, rather than exhibiting a clear peak. By dynamically changing the setpoint of the controller, other outflows patterns could be achieved.

The opportunity to set desired outflows and water levels based on management objectives will open up entirely management possibilities, which should be evaluated through future research. For example, control valves could be used to mimic "pre-development" conditions, which is often the goal of many stormwater infrastructure projects. Furthermore, rather than operating the system in a one-size-fits-all configuration, valves could be controlled based on multi-objective management goals. For example, the system could be operated for water quality benefits during smaller, more frequent storms, and operated for flood control during large storms. This further highlights the flexibility of real-time controlled systems, as their operation can adapt with changing watershed-level management goals, something that is difficult to accomplish using passive infrastructure.

5.8.3 Number of control points

As expected, the addition of each subsequent valve improves the performance of the control system. This is intuitive, since each control point provides additional dynamic storage and flexibility to buffer flows. More importantly, however, the benefit of successively adding valves is marginal, whereby adding more valves may not improve performance significantly. Economically, this is important, since it suggests that the entire catchment may not need to be controlled, but rather that simulation and engineering judgment can be used to determine the number of required control valves for a catchment. The number of required valves could then be chosen based on a specific management goal, or by finding the point at which investment into more valves will not provide significant returns.

Ultimately, each stormwater system has performance limits, which are a function of the hydrology, infrastructure, control objectives and costs, as well as the specific control algorithms. Real-time control will only be able to push water system to a certain point, beyond which new infrastructure construction may need to be considered. As illustrated in our own study, if construction is needed, new sites can be significantly smaller when real-time control is used. This is particularly important in many urban areas, where cost of construction is high and land availability may be limited.

In addition to having a sufficient number of control points, it is also important to determine where to place valves to maximize catchment-wide benefits. Given the lack of prior studies on this topic, our approach exhaustively simulated every possible configuration of valves across the entire catchment, which required over 15,000 model runs. For the catchment studied in this c, the locations selected for control could be prioritized based on their individual performance. For our study catchment, this means that the simulations of only one valve at a time (all other sites uncontrolled) could be used to rank sites, after which multi-valve configurations could be made by combining valves that had the best individual performance. As such, the number of simulations required for valve selection may only need to be high as the number of candidate sites in a catchment, which may significantly speed up future analyses for site selection. Practically, rather than requiring a specific configuration for a given number of valves, valves could be added without needing to change the location of the valve placed before it. This is very important, since valves can be added one-by-one to benefit the overall system, rather than requiring a pre-set configuration. Beyond exhaustive simulation, theoretical placement approaches (e.g. [214]) should also be evaluated, but they will need to be adapted to the unique dynamics of stormwater systems.

The physical characteristics of what makes one site more suitable for control than another are still not very clear. In our study, most of the ponds that ranked the highest in their ability to improve catchment-wide performance had a relatively low catchment area to volume ratio. In other words, they received very limited local runoff, but had large storage volume. This made them relatively suitable for buffering flows from upstream sites. As such, in-line storage may be a big factor in the ability of a site to contribute to catchment-wide benefits. However, this was not necessarily true of all “good” control sites and instead may be dependent on the actual catchment being studied. As such, more studies will be required in the future to determine if this is a reliable feature when selecting new control sites.

It did become clear in our study, however, that there are types of sites that may lead to worse performance compared to the uncontrolled case. This was particularly evident for SN5 and SN9, which overtopped when controlled. This occurred because the site had a small storage volume but large contributing runoff area, which did not permit it to react to rapid changes in runoff. For such storage nodes, feedback control should likely not be applied unless the cost function is adjusted for more conservative outcomes. This may also be overcome by predictive control, which will not only respond to real-time states, but also to forecasts for weather and runoff. Given the performance of the LQR-based approach, however, the application of model predictive control [182-184] of urban hydrologic catchment now appears very promising and will be evaluated in future studies. In fact, the role of weather uncertainty remains unstudied in the emerging field of real-time stormwater control and poses a promising research frontier.

This study serves as a baseline for assessing the integration of active control measures into urban catchments using dynamical control. While the approach shows great promise, several limiting assumptions were made that will need to be addressed in future studies. In all simulations, the storage nodes were initially empty and some remained filled after a storm. Once the storm had passed, nodes could be slowly drained to meet the outflow constraint. While this is not possible in passive systems, it may also become a problem if another storm begins before the storage nodes have been entirely drained. Alternatively, some catchments, such as those in the dry regions of the world may be configured for stormwater capture and reuse. In those cases, keeping storage nodes full becomes an objective, which poses risks to the control of the system if not enough storage is available to buffer incoming storms. This, again, stresses the importance of weather forecasts and their inherent uncertainty, which will need to be studied to determine how a system can be prepared ahead of incoming storms.

Given the nascent nature of real-time control, there is an urgent need to develop a framework to compare controlled and uncontrolled catchments on an equal footing. While it may be tempting to showcase plots of controlled hydrographs, the number of plots can quickly balloon, even for small systems. The cost functions that are used to parameterize control algorithms do not underpin the language used by decision makers and may ineffectively communicate the benefits to be gained by real-time control. To that end, an equivalence analysis will be necessary to contextualize and synthesize these benefits in terms of traditional systems. We believe that visualizations, such as those in (Figure 5.8) will provide a baseline intuition that can be used promote adoption.

Further, our centralized LQ controller assumed full knowledge of all states and zero noise, but the impacts of input and measurement uncertainty remain to be investigated. While modern sensors are becoming much more reliable and accurate, the role of sensor placement, measurement uncertainty, and sensor reliability must be studied to ensure robust control performance. Given the dynamical formulation of our framework, this could be accomplished through formal estimation approaches (e.g. Kalman filters [215]).

5.9 Conclusion

In the era of the self-driving car and smart energy grids, active control stands to transform the management of urban watersheds. We introduce a novel framework for analyzing the impact of real-time control across urban headwater catchments. By confirming the ability of feedback control to achieve desired flows and reduce flooding, the approach offers an alternative to new construction, which is currently the only solution to cope with changes in landuse and weather. The approach would, of course, need to be evaluated in the real-world, however, this should now be very feasible given that the necessary sensing and control technologies have been developed. The retrofitting of catchments should also be aided by the discovery that only a few key locations may need to be controlled, but this should still be validated on catchment-by-catchment basis.

Much research remains to be conducted to determine the generalizability and scalability of the methods proposed in this chapter. In particular, the control of larger urban watersheds should be evaluated. The authors contend that control at this larger scale may be most effectively achieved through the control of many smaller catchment “building blocks”. The need to segment control into smaller clusters may also be motivated by the practicality of working across ownership boundaries, insurance requirements, and social constraints. Social factors may ultimately become the most important barrier to the adoption of real-time control, since the best control algorithms may only be as good as the willingness of the public to adopt them. As such, there will be many opportunities to engage other disciplines in the emerging area of research. To promote transparency and accelerate future research of these topics, all of the models and source code from this study has been made available as part of an open source effort. Those interested in applying or contributing to these efforts are encouraged to join this web portal (open-storm.org).

Chapter 6
On the importance of weather forecasts in the control of stormwater systems

6.2 Introduction

As alluded to in earlier chapters, real-time control of stormwater infrastructure stands to drastically expand the possibilities for urban watershed management [38, 42]. While the results of the prior chapter show great promise, they also highlight a number of limitations that must be addressed. In particular, it was shown that in some instances, real-time control could lead to worse site-level outcomes when compared to static solutions. This may occur when aggressively-tuned cost functions push a site to an unsafe state, beyond which it cannot recover if overwhelmed by large or sudden inputs. Specifically, it was shown that controlling outflows requires the closure of valves, which, in turn, increases the amount of stored water. If a storm were to change abruptly and a site was already close to its storage capacity, local flooding may be inevitable if a valve could not be opened in time.

Forecasting poses a serious limitation for many stormwater applications, in particular for those in "dry" regions of the country. For example, in the city of Los Angeles, it is estimated that a single 1-inch summer storm may deliver as much as ten billion gallons of water [216]. Capturing this water would help provide direly needed water for this drought-stricken region. Unlike in the prior chapter, this would require that the system remain as "full" as possible, rather than draining during and after the storm. As illustrated, however, keeping a storage node full comes at the risk of local flooding if inflows change too rapidly. Releasing water in anticipation of a rain even may also be problematic, however. If a storm delivers less water than was anticipated this could lead to wasted water that could otherwise been held in the system. As such, there is a practical need to maximize captured stormwater without risking local flooding (Figure 6.1).

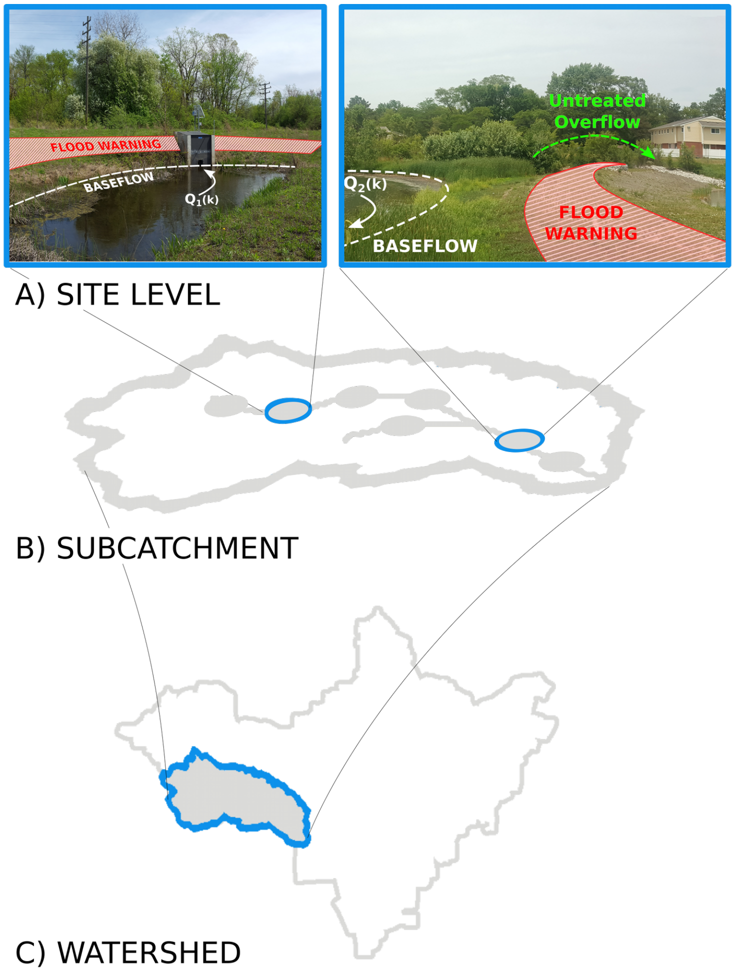


Figure 6.1. Capturing water at the scale of individual sites without causing flooding requires predictions across the scale of the entire watershed.

Since capacity cannot be created instantaneously, excess volume must be released in a timely fashion to lower the risks of local flooding and downstream erosion. In these cases, it would seem that the ability to forecast future states would improve performance by allowing the system to respond proactively rather than reactively. While forecasting will be critical to anticipating sudden changes in system states, weather patterns remain challenging to predict, particularly during large storm events. Not only is there variability in total rainfall, but the rainfall intensity also evolves over the duration of the storm event, as does the accuracy of the forecast (Figure 6.2). To the best of our knowledge, no prior studies have addressed these challenges in the context of stormwater control. Bridging this knowledge gaps stands to transform how far we can *push* stormwater systems without compromising the safety of inhabitants and property.

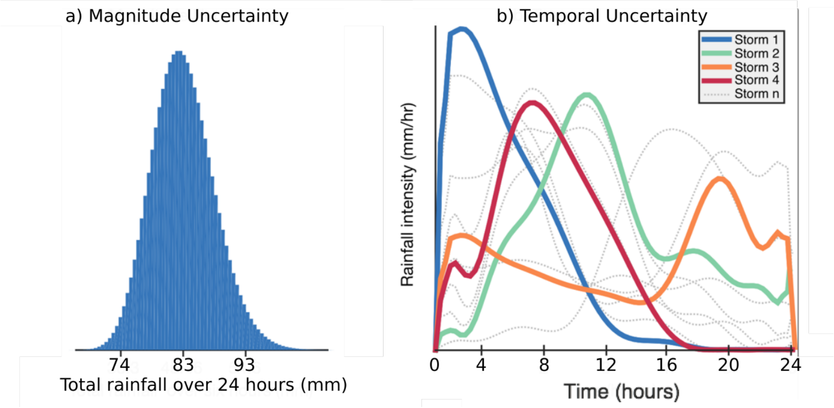


Figure 6.2. Sources of uncertainty include a) magnitude of the total rainfall during an event and b) the temporal uncertainty. In reality, the same weather forecast for one storm may manifest itself in any of these final outcomes.

In this chapter, we investigate the extent at which knowledge of future states improves control performance. Specifically, we ask the questions: (1) how should a stormwater system be prepared ahead of a storm to improve performance compared a feedback-only control approach, and (2) if forecasts are needed, how much of a warning window is enough to achieve desired performance (e.g. minutes, hours, days) and how accurate does the forecast need to be?

6.3 Case study

We revisit the model of a 0.83 km^2 urban catchment used in the previous chapter. However, this time we augment the control objective to meet the outflow objectives of the prior chapter while also keeping each storage node as full as possible to capture water for reuse. The highly impervious, urban subcatchment consists of a complex network of eleven storage nodes varying in volume from 370 m^3 to 32000 m^3 and overflow height set at 2.75 m. Each storage node was fitted with a controllable 0.1 m^2 square orifice located at the bottom of the node. Each orifice drained into a circular orifice, which varied in length from 40 to 360 m, each with a maximum depth of 0.275 m with a roughness of 0.01.

As an illustrative example, if the system is initialized at capacity (all storage nodes at 2.75m height) and controlled using the LQR approach from the previous chapter, flooding occurs because the system cannot respond rapidly enough to a storm (middle row of plots in Figure 6.3). However, if the storage nodes are drained partially before a storm, enough capacity is created to buffer incoming runoff while simultaneously ending up at the desired storage heights (bottom row of plots in Figure 6.3). In this chapter, we formalize how this can be accomplished.

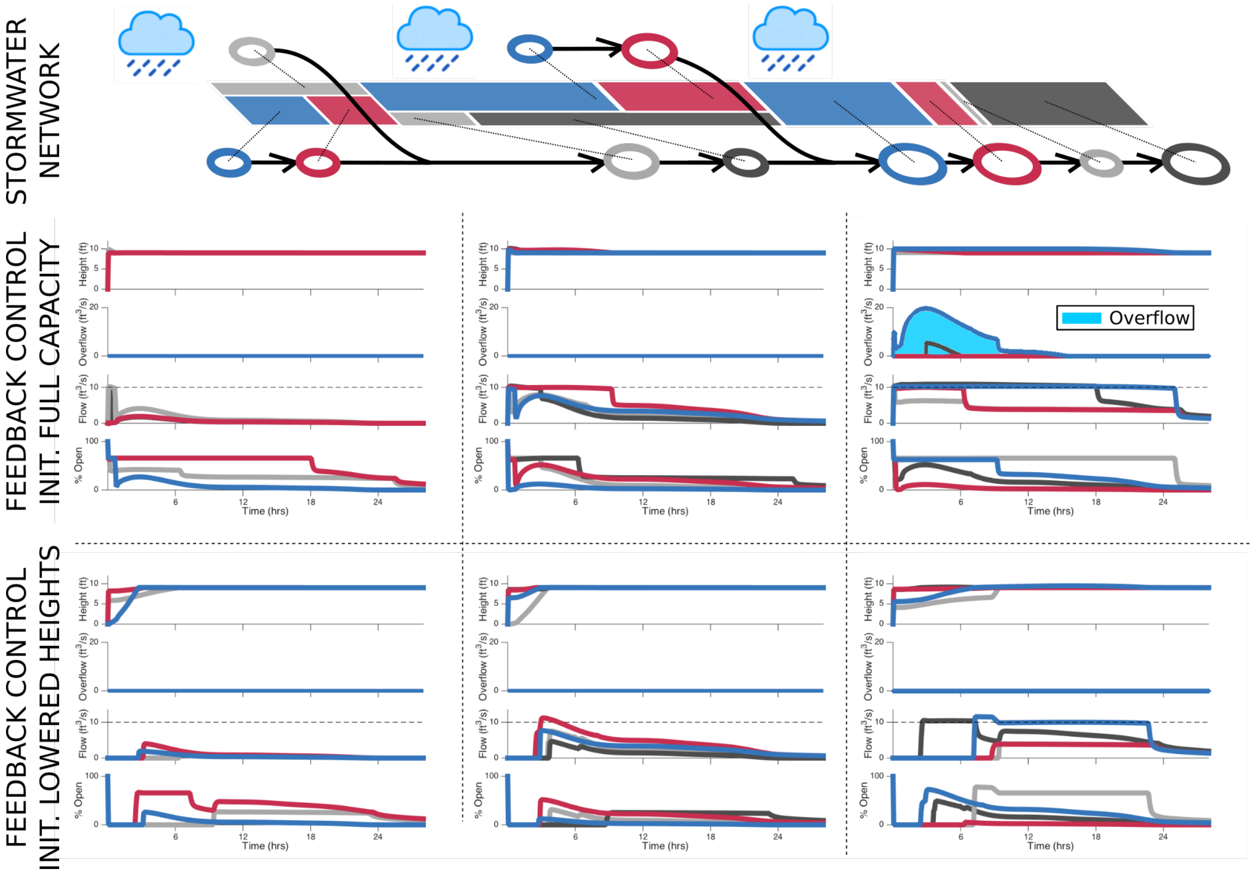


Figure 6.3. Network representation of the study catchment, with the fraction of the contributing subcatchment delineated for each storage node. System response with feedback control (middle row), and the benefit of creating storage ahead of a storm to avoid flooding (bottom row).

6.4 Methods

Here, we compare two control approaches that rely on the linear dynamics described in the prior chapter. The first approach provides a baseline case. It is a setpoint-based LQR controller, similar to that of the prior chapter. It starts with the storage nodes at full height and attempts to keep the nodes as full as possible during storm events. This may lead to flooding, however, if the system cannot adapt quickly enough in response to sudden inflows. The alternative approach relies on forecasting. It still uses the LQR controller, but prepares the starting heights based on a forecasted weather (drains the storage nodes before a storm).

6.4.2 Linear representation

In the previous chapter, an integrator-delay model was shown to be suitable for existing control methods and was used to produce a linearized state-space representation of the stormwater network (6.1). This state-space representation is also well suited for methods used for systems level analysis and control, such as reachability analysis, linear quadratic control, and model predictive control. The model is based upon physical dimensions of the network (e.g. ponded surface area, canal length, etc.) and was constructed from a SWMM model using the same approach outlined in Section 5.4.6.

$$\begin{gathered} x(k+1) = \mathbf{A}\,x(k) + \mathbf{B_u}\,u(k) + \mathbf{B_d}\,d(k) \\ y(k) = \mathbf{C}\,x(k) \end{gathered} \tag{6.1}$$

This study assumed a time-invariant system, as well as full knowledge of all states with zero noise. The state vector $x(k)$ is a 66 x 1 element vector consisting of 11 storage elements and a total of 55 flow delay terms; $u(k)$ is an 11 x 1 vector of the control out flows for each storage node; d is a 19 x 1 vector of the storm water runoff from each subcatchment. $\mathbf{A}$ is a 66 x 66 matrix, $\mathbf{B_u}$ is a 66 x 11 matrix, and $\mathbf{B_u}$ is a 66 x 19 matrix.

6.4.3 Modified Feedback Control

Unlike in the prior chapter, where LQR-control was used to drive the system toward a zero-state, a modification has to be made to allow non-zero set points. This is important to allow storage nodes to be controlled to specific heights. LQR can be formulated to track a given reference [217, 218]. Given a cost function J:

$$J = x^T(N)\,\mathbf{Q}\,x(N) + \sum_{k=0}^{N-1}(\rho \cdot [\mathbf{H}\,x(k) - h^*\,]^T\,\mathbf{Q}\,[\mathbf{H}\,x(k) - h^*\,] + u^T(k)\,\mathbf{R}\,u(k)) \tag{6.2}$$

Where **H** is the setpoint matrix is used to extract the states that are to be controlled. In this case, the matrix **H** is an 11 x 66 matrix that extracts the height of each storage node from $x(k)$. The control outflow is given by the vector $u(k)$:

$$u(k) = -\mathbf{K}\,x(k) + \mathbf{E}\,h^* \tag{6.3}$$

where:

$$\mathbf{Q} = \mathbf{Q^T} \geq \mathbf{0}, \mathbf{R} = \mathbf{R^T} > \mathbf{0}, \tag{6.4}$$

$$\mathbf{K} = \left(\mathbf{R} + \mathbf{B_u}^T\mathbf{P}\mathbf{B_u}\right)^{-1}\mathbf{B_u}^T\mathbf{PA} \tag{6.5}$$

$$\mathbf{E} = (\mathbf{R} + \mathbf{B}_\mathbf{u}^T\mathbf{PB})^{-1}\mathbf{B}_\mathbf{u}^T\mathbf{T} \tag{6.6}$$

and the matrix $\mathbf{P} = \mathbf{P}^T \geq \mathbf{0}$ is the solution to the Discrete Time Ricatti Equation (6.7):

$$\mathbf{A^TPA} - \mathbf{P} - \left(\mathbf{A^TP}\,\mathbf{B}_\mathbf{U}^\mathbf{T}(k)\right)(\mathbf{B}_\mathbf{u}^\mathbf{T}\mathbf{P}\,\mathbf{B_u} + \mathbf{R})^{-1}(\mathbf{B}_\mathbf{u}^\mathbf{T}\mathbf{P}\,\mathbf{A}) + \mathbf{H}^T\mathbf{QH} = 0 \tag{6.7}$$

Solving for **P** can be accomplished using the solution from the prior chapter in Section 5.4.7. To solve for **T**:

$$\mathbf{T} = -[\mathbf{A} - \mathbf{A}^T\mathbf{PB_u}(\mathbf{R} + \mathbf{B}_\mathbf{u}^T\mathbf{PB})^{-1}\mathbf{B}_\mathbf{u}^T - \mathbf{1}_{nxn}]^{-1}\mathbf{Q} \tag{6.8}$$

6.4.4 Preparatory LQR (pLQR) Feedback Control

The alternative approach addresses the limitations of feedback control by adding a preparatory step. Given a forecasted storm disturbance it calculates the starting heights of the storage nodes ahead of time, with the goal of creating enough storage to capture incoming flows without flooding. In simpler terms, it estimates how much water will be delivered by a storm and then removes just that much volume from the storage nodes before the storm begins. This can be accomplished deterministically across entire systems by inverting the linear dynamics and back-calculating a best estimate of initial state for a given forecast (6.15). Similar approaches have been studied before in the context of scheduling for irrigation canals using both heuristic algorithms and by inverting the solution for unsteady, open-channel flow [219, 220]. By induction, we can deterministically express the state at timestep, k, using (6.9):

$$x(k) = x_0(k) + x_u(k) + x_d(k) \tag{6.9}$$

$$x_0(k) = \mathbf{A}^k x(0) \tag{6.10}$$

$$x_u(k) = \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\, \mathbf{B_u}\, u(j) \tag{6.11}$$

$$x_d(k) = \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\, \mathbf{B_d}\, d(j) \tag{6.12}$$

Assuming a simulation has N timesteps, $x_d(N)$ represents the total change in height by the end of the simulation. Disregarding the control input and focusing only on the effects of runoff, the estimated initial state $\hat{x}\,(0)$ to reach the desired final state $x(N)$ is approximated by (6.15):

$$x(N) = \mathbf{A}^N\, \hat{x}\,(0) + \sum_{j=0}^{N-1} \mathbf{A}^{(k-1-j)}\, \mathbf{B_d}\, d(j) \tag{6.13}$$

$$\mathbf{A}^N\, \hat{x}\,(0) = x(N) - \sum_{j=0}^{N-1} \mathbf{A}^{(k-1-j)}\, \mathbf{B_d}\, d(j) \tag{6.14}$$

$$\hat{x}\,(0) = \mathbf{A}^{-N}\left(x(N) - \sum_{j=0}^{N-1} \mathbf{A}^{(k-1-j)}\,\mathbf{B_d}\,d(j)\right) \tag{6.15}$$

In practice, since $\hat{x}\,(0)$ represents the heights of each storage node and must be non-negative, the implemented initial state x(0) are given by the vector (6.16):

$$x(0), \text{ where } x_i(0) = max(\hat{x}_i\,(0),\, 0) \tag{6.16}$$

Given a “perfect” storm forecast and assuming that the linear dynamics fully describe the system, this approach would require little to no feedback control, as the initial state could theoretically always be driven to the desired final state by the rainfall alone. This is, however, unrealistic since storms patterns are uncertain and the linearized system does not capture the complete dynamics. As such, our control approach takes the initial state but still applies feedback control to remedy unforeseen future states. Since LQR is used in this study as the feedback controller, the approach will hereby be referred to preparatory LQR, or pLQR. While beyond the scope of this chapter, the process to obtaining the initial state $x(0)$ can be extended further (please see Appendix 7.2A.5.2) to evaluate what the controlled outflows should be in order to reach the desired final state when starting from $x(0)$.

6.4.5 Simulating Weather

The two approaches were evaluated across a series of synthetic storm events. To simulate forecast uncertainty for the synthetic events, a stochastic sampling approach was developed using designs storms. For storm synthesis, this study used regionalized statistical data provided online[14] from *Precipitation-Frequency Atlas 14 of the United States* [206]. Point estimates of the average intensity with upper and lower limits of the 90% confidence interval were used to generate a distribution for rainfall intensity for 10-year, 24-hour storm events. In the case of Ann Arbor, Michigan, based upon the regional data for a 10-year, 24-hour storm, the average was 83 mm with lower and upper bounds of 74 mm and 93 mm, respectively (Figure 6.4). Rainfall intensity was modeled using a gamma distribution, one of the most common models used for modeling rainfall distribution [221, 222], particularly for daily rainfall [223]. The gamma distribution was parameterized such that the mean and variance approximated both the average rainfall intensity and the 90% confidence interval, respectively. Using a gamma distribution also ensured all randomly generated precipitation values would be non-negative.

Atlas 14 also provides temporal distributions for heavy rainfall amounts [206]. The online database hosts regionalized statistical rainfall timeseries curves expressed as the average cumulative percentage of total precipitation. In addition to the average, four separate distribution curves were also available, sorted by the quartile within which the greatest amount of precipitation fell (Figure 6.4). For example, a second-quartile 24-hour storm would have the highest percentage of total rainfall occurring between the sixth and twelfth hours. The likelihood of each distribution for 24-hour storms in the North Plains region is 50%, 21%, 16%, and 13% for the first-, second-, third-, and fourth-quartiles, respectively[15]. Each quartile was refined into ten deciles for further temporal variation.

[14] Precipitation data frequency server http://hdsc.nws.noaa.gov/hdsc/pfds/

[15] Table A.5.1 - NOAA Atlas 14 Volume 8 Version 2.0

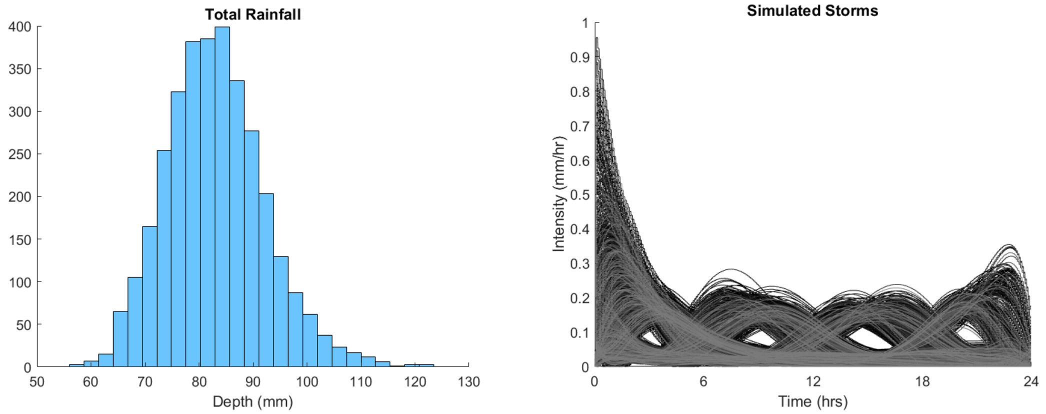


Figure 6.4. Rainfall and temporal variability of the 3300 synthesized storms.

When synthesizing a storm, the first step was to sample twice from a uniform distribution to determine the quartile and decile and generate a temporal curve with unit intensity. Each temporal curve was then fit with a tenth-order polynomial, per recommendation [206], and then scaled by a rainfall intensity that was randomly sampled from the gamma distribution. The scaled temporal curve was finally interpolated every five minutes to generate the timeseries for the storm event used in the simulation. Spatial variability was not considered as rainfall was assumed to be uniform across the entire subcatchment. In all, 3300 10-year, 24-hour storms were synthesized, where the average rainfall intensity was 83 mm with a standard deviation of 7.4 mm, agreeing with the point precipitation frequency estimates hosted by NOAA[14] (Figure 6.4). Of these storms, 49.7%, 21.9%, 15.9%, and 12.7% were first-, second-, third-, and fourth-quartile cases.

6.5 Implementation and Evaluation

6.5.2 Simulation Framework

A Matlab-based, open-source co-simulation framework was used to model the hydraulic and hydrologic dynamics, following the same steps as in the prior chapter. A total of 3300 rainfall timeseries were generated (Figure 6.4) at five-minute resolution to simulate design storms with varying intensities and temporal variability. The control inputs for both LQR and pLQR were computed at each timestep using (6.3). Simulations were run in Matlab using up to one hundred cores in parallel on a high-performance Linux computing cluster[16].

6.5.3 Performance Evaluation

The control approaches, LQR and pLQR, were compared across the synthetic design storms. For LQR, it was assumed that all storage nodes started at full capacity (2.75 m height) at the onset of the storm, whereas pLQR was initialized at the height calculated using relation (6.15). Specifically, the pLQR-controlled system was initialized based the average rainfall of 83 mm, the average of the synthesized rain events. Control cost functions were tuned to i) mitigate flooding, ii) capture stormwater runoff and iii) limit outflows beyond a flow threshold to minimize critical sheer stress for mobilizing solids and leading to stream erosion [207-209]. The recommended critical flow limit was 25% of 2-year peak flows [224]. The final performance metric in this study evaluated mainly flooding and stream erosion (outflows exceeding a critical threshold) since both control approaches always met the desired set point height of 2.75 m

[16] Flux http://arc-ts.umich.edu/systems-and-services/flux/

(Figure 6.3). Practically, the final comparison primarily evaluated how much flooding was reduced while still capturing the maximum amount of water.

6.5.4 Controller comparison

The performance each approach is presented in Figure 6.5. In both cases, the LQR and pLQR controllers were able to meet the final setpoint heights of all storage nodes within a respectable 0.05 m margin. As expected, however, pLQR improved on flood reduction compared to LQR due to more available storage at the onset of the storm. Overall, pLQR was able to reduce flood volumes by over 71%, on average, compared to LQR. In fact, in over 1030 instances, pLQR simulations saw no flooding at all, while flooding was evident in every LQR simulation. The pLQR controller also achieved improved outflow characteristics and greatly reduced the total time outflows exceeded the critical erosion threshold (by nearly 20% reduction on average). The resulting outflow distribution for pLQR had a near bi-modal shape with peaks near 16 and 28 hours, where the peak near the 28-hour mark is attributed to storms with intensities greater than 0.5 mm/hr. These large, flashy first-quartile storms made up 1135 of the 3300 stochastically generated storms (~33%).

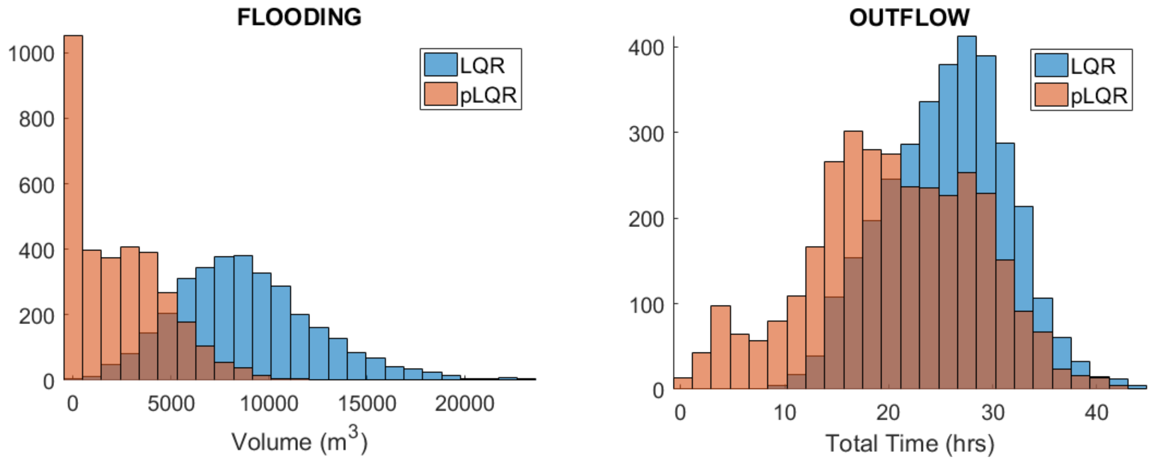


Figure 6.5. Histogram of total flooding volume and total time critical outflow levels were exceeded for all design storms using LQR and preparatory LQR (pLQR) controllers.

6.6 Exploring real-world constraints: a case study

The prior section illustrated the benefit of preparing a stormwater system in anticipation of a storm event. While this worked well for the series of synthetic storms, the pLQR method will be challenged by a number of real-world considerations. In particular, weather forecasts are often notoriously inaccurate across long time horizons. A real-world forecast is shown in Figure 6.6 for the city of Ann Arbor, illustrating that the daily forecast (green), differs distinctly from the hourly (yellow) and minute-by-minute (blue) forecast. As such, preparing the system a day ahead of time, without being certain about the magnitude and dynamics of the event, may compromise control performance. Secondly, and more importantly, many storms rarely occur on their own. Namely, one storm is often forecasted while another is still ongoing. As such, any notion of "pre-draining" the system is limited during long or multi-peak storms.

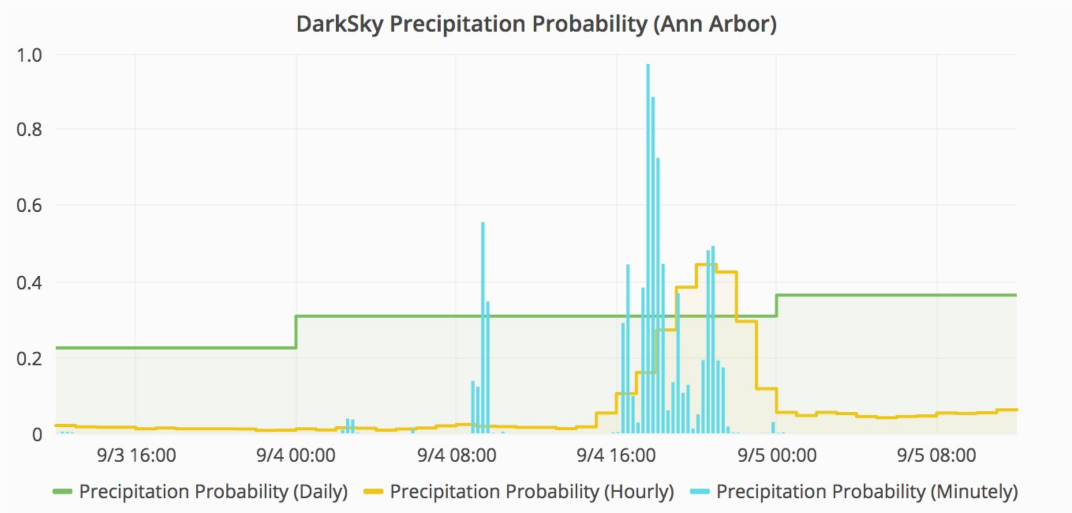


Figure 6.6. Evolution of precipitation forecasts over time, showing that the chance of rainfall differs between the daily, minute, and even on a minute-scale resolution.

Given that the value of predictive control has been verified, rather than adopting the pLQR method to these challenges it may be quicker, for the illustrative purposes of this final chapter, to simply apply more modern predictive controllers. To that end, we close this dissertation by applying model predictive controller (MPC) to the problem of controlling a system of stormwater ponds given a rainfall forecast. The use of MPC will permit for control decisions to be made continuously throughout the storm, while also considering future rain inputs. The performance of this approach will be evaluated using our own archived weather forecasts and recorded rainfall data for our study catchment[17]. Over the month of May 2017, forecasts were archived from Weather Underground every five minutes. The measured rainfall was also recorded and used as the true rainfall during the control simulations. Overall, thirteen storms of

[17] Weather Underground https://www.wunderground.com/

various size and duration were reported. May 3rd marked the largest event that lasted twenty-four hours with a cumulative rainfall of 21 mm and a temporal variability that included two peaks.

6.6.2 Model Predictive Control (MPC)

Model predictive control (MPC) has received much attention in process control systems [225, 226] and more recently in irrigation and canal systems [182, 183, 227, 228], however MPC has yet to be applied for system-level control of urban watersheds. MPC is a predictive approach, which computes a new control solution at each time step with the ability to account for forecasts of future inputs. This is unlike LQR, which uses a single (optimal) solution for the entire event [229]. While the controller generates a sequence of control inputs to optimize the performance over a potentially long-time window, typically only the first timestep is implemented and the process is repeated given new input forecasts. This makes MPC highly attractive for use in stormwater systems, since weather forecasts improve closer in time to the storm event. Unlike our second approach, pLQR, this means with MPC that smaller control actions can be taken using more recent forecasts, without needing release water well ahead of the storm.

Our implementation of MPC also relies on the same linear dynamics given by the integrator delay model and is subject to a cost function (6.17). The cost function penalizes deviation from desired storage height, outflows, and the change in outflow. MPC was implemented using a 3-hour prediction window with a 1-hour control horizon. Outflows were constrained to be between 0 and 0.29 m^3/s and the observed heights constrained between 0 and 2.75 meters. There was no constraint on the rate of change for the control outflow from any storage node. The weights for all outflows, change in outflow, and deviation from desired heights, h^*, were $w_Q = 1$, $w_{\Delta Q} = 0.1$, and $w_h = 3500$, respectively. The cost function at timestep k is given by:

$$P_k = \sum_{i=0}^{H} \big(w_Q \cdot [Q^T(k+i)\, Q(k+i)] + w_{\varDelta Q} \cdot [\varDelta Q^T(k+i)\, \varDelta Q(k+i)] + w_h \cdot [h(k+i) - h^*(k+i))^T (h(k+i) - h^*(k+i)] \big) \tag{6.17}$$

where P_k is the performance measure at timestep k, and w_x, w_u, $w_{\varDelta u}$ are weights and H is the number of timesteps in the prediction horizon.

6.6.3 Archived forecasts and evaluation

The MPC approach was compared to the feedback-only LQR approach by simulating the system during a series of real-world weather events and forecasts. MPC was evaluated using the original forecasts as well as the measured rainfall. The latter was intended to illustrate the case of a "perfect" weather forecast to provide an upper bound on performance. Throughout May 2017, a total of 62 mm of precipitation fell over 39 hours, resulting in approximately 41,000 m^3 in runoff throughout the study subcatchment (Figure 6.7). In particular, there was a 21-mm storm on May 4. Scaling the rainfall such that the May 4 storm was an 83 mm, 24-hour storm resulted in a total of 248 mm of precipitation over 39 hours and approximately 160,000 m^3 in runoff. For the full set of simulations, this required recurrently simulating the runoff over a 24-hour horizon with the updated rain forecast for each five-minute time step, resulting in over 8900 simulations during the 744-hour (1 month) time window. Furthermore, to simulate the impacts of larger storm events, the three cases (LQR, MPC with real forecast, and MPC with "perfect" forecast) were then repeated by scaling up the rainfall intensity such that the largest storm had an intensity of 82 mm over a 24-hour period, approximately that of a 10-year, 24-hour storm for the region[14].

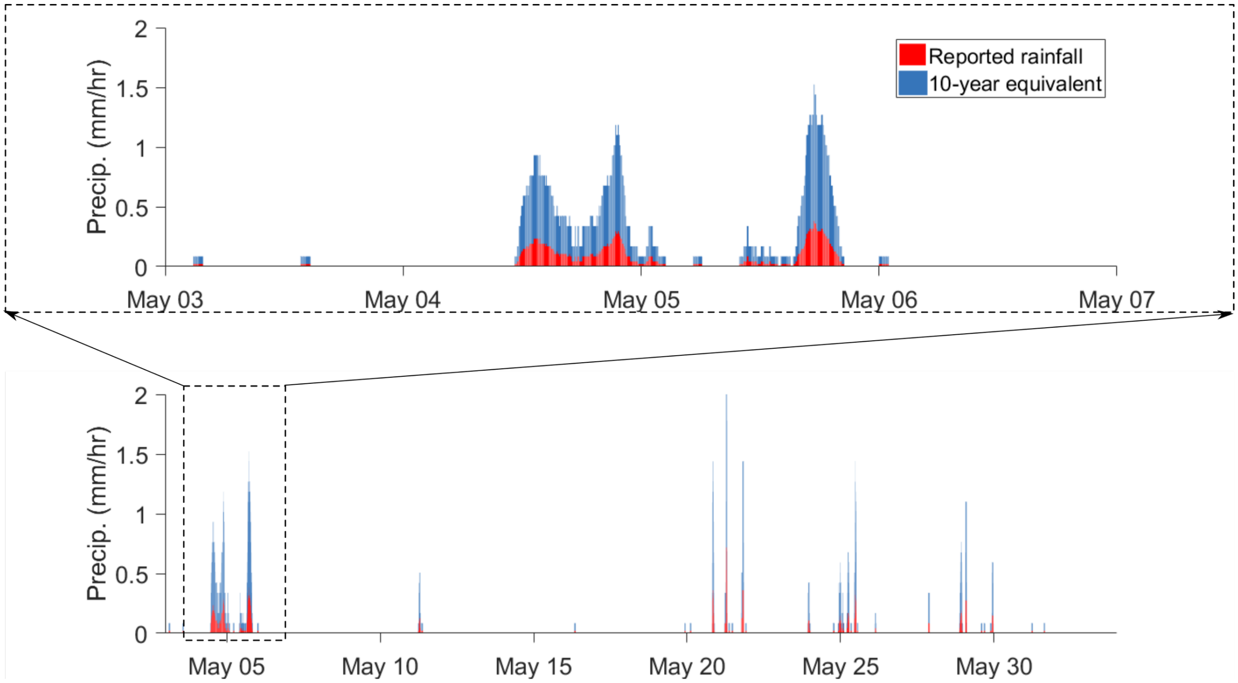


Figure 6.7. Recorded and scaled rainfall in Southeastern Michigan throughout May 2017.

6.6.4 MPC Performance

Both MPC and LQR were able to meet the desired setpoint heights (full storage nodes) across all simulations. For the scenario with the smaller set of storms in May of 2017, LQR and MPC showed nearly identical performance across all criteria (Table 6.1). Since the storms were relatively small, no flooding occurred in any of the cases, thus highlighting no discernable benefit of using forecasts for control. However, when scaling the precipitation up, the model predictive controller performed better across all categories (Table 6.2). The MPC approach showed a 10% reduction in flooding volumes when using the forecasts, and a 26% reduction in flooding when using a "perfect" forecast. As such, there is a significant benefit to using forecasts, but the performance is highly reliant on the quality of the weather forecast.

MAY 2017 (744 HR) SIMULATION RESULTS

	avg. final height (m)	time (hr) outflows > 0.29 m^3/s	volume (m^3) flooding
FEEDBACK	2.75	0.12	0.0
MPC REAL FORECAST	2.75	1.3	0.0
MPC PERFECT FORECAST	2.75	1.2	0.0

* 41 000 (40 873) cubic meters total runoff volume; total rainfall duration: 39 hours

Table 6.1. Simulation results for all storms throughout May 2017

MAY 2017 (744 HR) SIMULATION RESULTS

	avg. final height (m)	time (hr) outflows > 0.29 m^3/s	volume (m^3) flooding
FEEDBACK	2.75	8.3	19 600
MPC REAL FORECAST	2.75	6.9	17 400
MPC PERFECT FORECAST	2.75	7.7	15 400

* 170 000 (168 916) cubic meters total runoff volume; total rainfall duration: 39 hours

Table 6.2. Simulation results for storms throughout May 2017 up-scaled to 10-year storms. MPC outperforms the non-predictive approach reducing flooding as well as erosive outflows.

6.7 Takeaways on the role of weather

This chapter provided an introductory note on the role of weather uncertainty in the multi-objective control of stormwater systems, a topic which has little, if any, attention from the research community. Our analysis illustrated that there are clear benefits to predictive control approaches, especially during large storm events. The objectives in used in the example case study, which required all storage nodes to remain full, may seem extreme, but the results indicate that it may actually be possible to push stormwater systems towards these performance zones. As storms continue to grow in frequency, intensity, and duration, this opens the opportunity to begin using real-time control for multiple purposes including stormwater capture, without compromising safety of the public and nearby property through flooding and pollution.

Much future work remains to be conducted on this topic however. Better weather forecasts clearly improve the performance of predictive controllers, but the controllers will also ultimately be limited by their formulation and assumption of linear dynamics. As such, it is still unclear if robust control algorithms are more important than better weather forecasts. Furthermore, non-uniform rainfall patterns and their associated uncertainty should also be evaluated in future studies.

Chapter 7 Conclusion

7.1 Summary of Discoveries

The goal of this dissertation was to enable a foundation for the real-time study and control of urban watersheds. To that end, the specific contributions tackled a variety theoretical and technological challenges, which ultimately led to a number of fundamental conclusions:

- **Chapter 2**: We illustrated that real-time controlled stormwater systems have great potential to transform how we manage flows and water quality in urban watersheds.
- **Chapter 3**: We learned that web-services and wireless connectivity offer scalable and reliable means to build and deploy large water sensors networks.
- **Chapter 4**: We discovered that water quality dynamics are site-specific and vary across scales, thus calling for more dynamic management approaches, such as those offered by real-time control. We also illustrated how this could be achieved using an unprecedented real-world study, in which a watershed was controlled using valves.
- **Chapter 5**: We learned that linear-feedback control has the potential to drastically change the flows in urban headwater catchments and illustrated that only a relatively small number of sites have to be controlled to achieve these benefits.
- **Chapter 6**: We discovered that the use of weather forecasts stands to play a large role in improving control performance, especially for scenarios that would otherwise be too risky with non-predictive approaches.

7.2 Future directions

Due to its modular nature, the web-based framework introduced in Chapter 3 can be expanded to use more robust platforms and technologies as they emerge. This includes web platforms (e.g., CHORDS, InfluxDB [154] and Amazon IoT [155]), sensors (e.g., water quality and air quality), and wireless technologies (e.g., Wi-Fi, Bluetooth, and mesh networks). The framework has already been used in other environmental sampling applications, such as adaptively sampling lake conditions for hypoxic regions that may be indicative of harmful algal blooms [230], and can be readily scaled up and ported to other sensing and control applications.

The adaptive sampling algorithm from Chapter 4 could be extended to incorporate real-time stormwater modeling by combining it with the co-simulation framework from Chapters 5 and 6 to better forecast and estimate flows throughout the catchment. Lessons from these controller developments can be applied to develop more complex sampling schemes that integrate upstream sensor data and forecasts. As the network of autosamplers is expanded, they could be deployed to capture spatial variability throughout a watershed, as well as configured to sample around valves to capture the effects of real-time control on water quality.

In chapter 5, the goal to identify the "best" sites for control may have yielded results that were site specific. It remains to be determined what can be generalized so that exhaustive simulations may no longer be necessary to pinpoint which sites should be retrofitted for real-time control. After investigating the role of weather uncertainty in Chapter 6, better forecasting techniques must be developed to estimate the peak rainfall intensity and cumulative rainfall of a storm event. It remains to be evaluated if this is still valid for storms of different durations and intensities, such as 2-year, 6-hour storms up to 100-year, 96-hour storms. The spatial variability in rainfall also poses an unstudied challenge in systems-level stormwater control. The controllers in Chapter 5 and Chapter 6 assumed perfect state information and that the system was fully observable. This may not always be the case due to sensor noise compounded with outages in communication and sensing technologies. The integration of data assimilation and techniques, such as Kalman filtering, is much needed and remains to be explored.

Since its inception, Open-storm.org has grown into a diverse community of all those interested in sensing for stormwater. With quick access to the latest technologies, methods, case studies, and results, developers and researchers can now more easily engage decision-makers and stakeholders. To broaden the exposure of real-time storm water measurement and control, Open-storm should also be expanded as an educational tool for secondary schools and used as a means for neighborhood involvement and community outreach.

This thesis has only begun to tackle the challenges in the nascent field of “smart” stormwater systems. While many technological and theoretical challenges remain, there are also many social barriers that will determine if these systems will be adopted. Economics and the social sciences will play a vital role in convincing the public and decision makers of the value and safety of “smart” systems. This promises to expand this already multidisciplinary field into an even broader research community. As the achievements in this thesis were made possible by integrating other disciplines to solve problems in stormwater, by extension, it is reasonable to assume these advances are applicable to other fields of work, including but not limited to agricultural fields and beyond.

Appendix

A.1 Chapter 3 Software and data availability

The use case in Chapter 3 was implemented using the Xively Internet of Things platform and a Flask web-server (written in Python 2.7) running on an Elastic Beanstalk t2.micro instance provided by Amazon Web Services. All experimental data from the study are hosted on a secure Xively feed and available upon request. The source code and implementation parameters are available on a public repository: https://github.com/kLabUM/IoT. As of 2015, all of these tools are available at no cost for a project of the scale discussed in this chapter. Web connectivity is required of all hardware and software.

A.2 Chapter 4 Software, Hardware, and Data Availability

Data and source code, as well as hardware and sensor schematics used to produce the figures in this chapter can be obtained by directly contacting the authors or visiting their website (http://www.tinyurl.com/bkerkez).

A.3 Chapter 5 Resource Availability

To promote transparency, all resources from this chapter, including the physical simulation framework, control algorithms, and example models, are available as an open source package on http://github.com/open-storm. All interested parties are invited to replicate our analyses or adapt the toolbox to their own control of water systems.

A.4 State-space model of the study catchment

State-space representation of the study catchment as a discrete, time-variant system is defined by the state vector, $x(k)$ – which is composed of the heights, $h_i(k)$, and flows $Q_i(k)$ of all storage nodes in the system at the k^{th} timestep; the control vector, $u(k)$, contains the control outflow $Q_{control,\ i}(k)$ for each controllable orifice/valve; and the disturbance vector, $d(k)$, is a vector of the total runoff $Q_{runoff,\ i}(k)$ from all local subcatchments that flow into the i^{th} storage node. The state matrix, $\boldsymbol{A}(k)$ relates the heights and flows from the current timestep to the next; the control matrix, $\boldsymbol{B_u}(k)$, links the control outflow to its associated storage node; and the disturbance matrix, $\boldsymbol{B_d}$, routes the rainfall-generated runoff from a given subcatchment to its associated storage node.

$$x(k+1) = \mathbf{A}(\mathrm{k})\, x(k) + \mathbf{B_u}(k)\, u(k) + \mathbf{B_d}\, d(k)$$
$$y(k) = \mathbf{C}\, x(k)$$

An example of the time-invariant case where all eleven storage nodes are controllable is presented below. The non-zero elements of each matrix are plotted and the values are presented in sparse format.

Additional details can be found in Sections 5.4.4 and 5.5.2.

A.4.1 State Matrix, A

The non-zero (nz) elements of the state matrix, $\boldsymbol{A}$, are plotted and reproduced below:

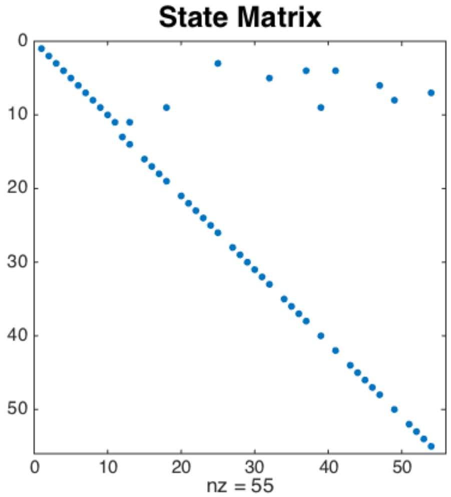


Figure A.1. The non-zero elements of the state matrix, $\boldsymbol{A}$.

(1,1)	1
(2,2)	1
(3,3)	1
(4,4)	1
(5,5)	1
(6,6)	1
(7,7)	1
(8,8)	1
(9,9)	1
(10,10)	1
(11,11)	1
(13,12)	1
(11,13)	0.0270
(14,13)	1
(16,15)	1
(17,16)	1
(18,17)	1
(9,18)	0.0175
(19,18)	1
(21,20)	1
(22,21)	1
(23,22)	1
(24,23)	1
(25,24)	1
(3,25)	0.1324
(26,25)	1
(28,27)	1
(29,28)	1
(30,29)	1
(31,30)	1
(32,31)	1
(5,32)	0.0105
(33,32)	1
(35,34)	1
(36,35)	1
(37,36)	1
(4,37)	0.0048
(38,37)	1
(9,39)	0.0175
(40,39)	1
(4,41)	0.0048
(42,41)	1
(44,43)	1
(45,44)	1
(46,45)	1
(47,46)	1
(6,47)	0.0050
(48,47)	1
(8,49)	0.2040
(50,49)	1
(52,51)	1
(53,52)	1
(54,53)	1
(7,54)	0.0044
(55,54)	1

A.4.2 Control Matrix, B_u

The non-zero (nz) elements of the disturbance matrix, $\boldsymbol{B_u}$, are plotted and reproduced below:

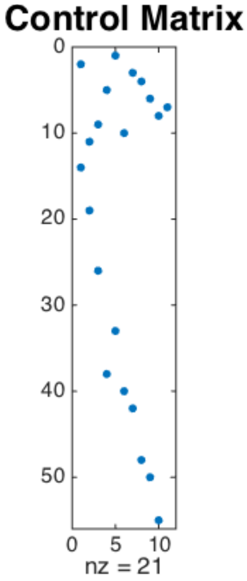


Figure A.2. The non-zero elements of the control matrix, $\boldsymbol{B_u}$.

(2,1)	0.1520
(14,1)	-0.3605
(11,2)	0.0097
(19,2)	-0.3605
(9,3)	0.0063
(26,3)	-0.3605
(5,4)	0.0038
(38,4)	-0.3605
(1,5)	0.0108
(33,5)	-0.3605
(10,6)	0.0258
(40,6)	-0.3605
(3,7)	0.0477
(42,7)	-0.3605
(4,8)	0.0017
(48,8)	-0.3605
(6,9)	0.0018
(50,9)	-0.3605
(8,10)	0.0735
(55,10)	-0.3605
(7,11)	0.0016

A.4.3 Disturbance Matrix, B_d

The non-zero elements (nz) of the disturbance matrix, $\boldsymbol{B_d}$, are plotted and reproduced below:

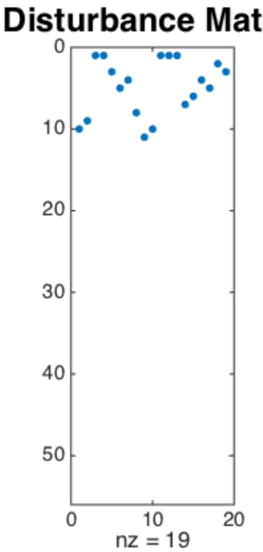


Figure A.3. The non-zero elements of the disturbance matrix, $\boldsymbol{B_d}$.

(10,1)	0.0716	(8,8)	0.2040	(6,15)	0.0050
(9,2)	0.0175	(11,9)	0.0270	(4,16)	0.0048
(1,3)	0.0300	(10,10)	0.0716	(5,17)	0.0105
(1,4)	0.0300	(1,11)	0.0300	(2,18)	0.4216
(3,5)	0.1324	(1,12)	0.0300	(3,19)	0.1324
(5,6)	0.0105	(1,13)	0.0300		
(4,7)	0.0048	(7,14)	0.0044		

A.4.4 Output Matrix, C

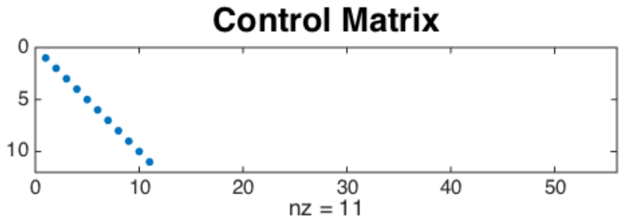


Figure A.4. The non-zero elements of the output matrix, $\boldsymbol{C}$

(1,1)	1	(5,5)	1	(9,9)	1
(2,2)	1	(6,6)	1	(10,10)	1
(3,3)	1	(7,7)	1	(11,11)	1
(4,4)	1	(8,8)	1		

A.5 Supplementary Information for Chapter 6

A.5.1 Regional precipitation data

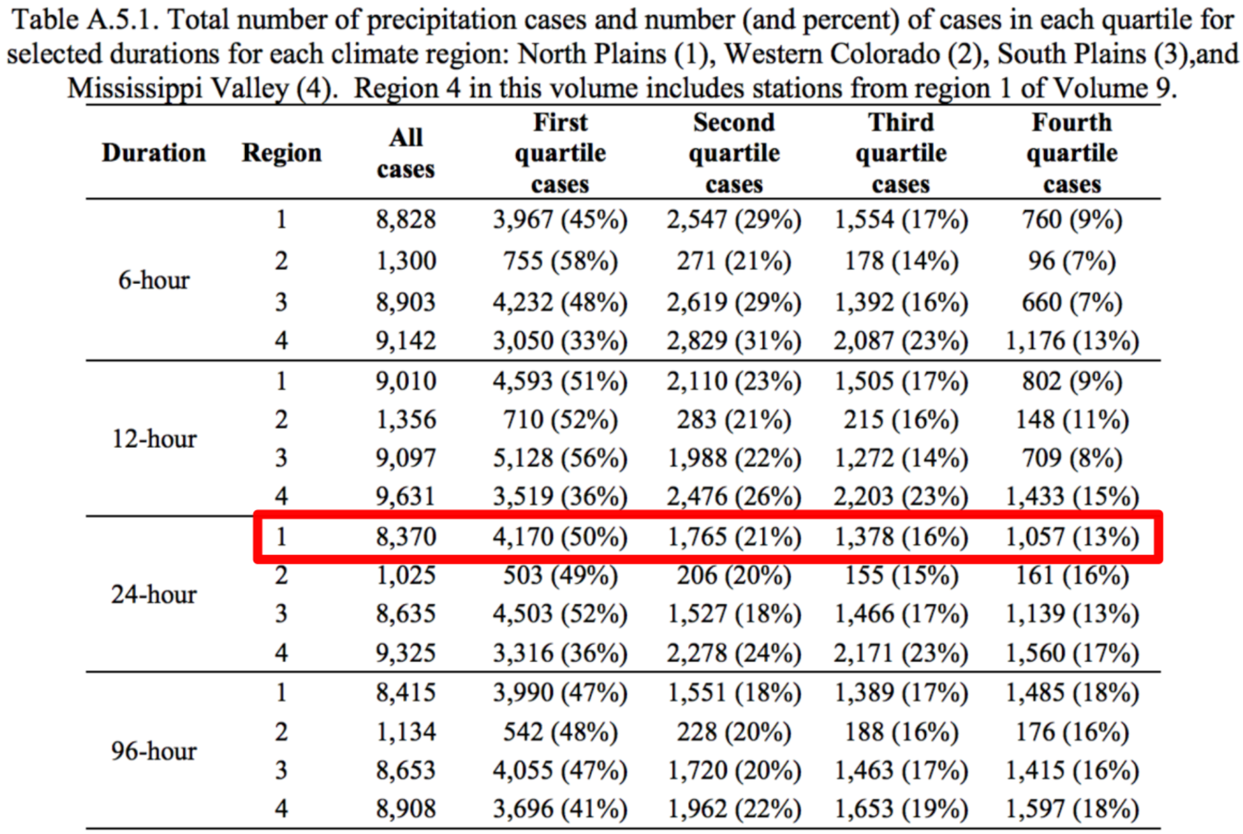


Table A.5.1. Total number of precipitation cases and number (and percent) of cases in each quartile for selected durations for each climate region: North Plains (1), Western Colorado (2), South Plains (3),and Mississippi Valley (4). Region 4 in this volume includes stations from region 1 of Volume 9.

Duration	Region	All cases	First quartile cases	Second quartile cases	Third quartile cases	Fourth quartile cases
6-hour	1	8,828	3,967 (45%)	2,547 (29%)	1,554 (17%)	760 (9%)
	2	1,300	755 (58%)	271 (21%)	178 (14%)	96 (7%)
	3	8,903	4,232 (48%)	2,619 (29%)	1,392 (16%)	660 (7%)
	4	9,142	3,050 (33%)	2,829 (31%)	2,087 (23%)	1,176 (13%)
12-hour	1	9,010	4,593 (51%)	2,110 (23%)	1,505 (17%)	802 (9%)
	2	1,356	710 (52%)	283 (21%)	215 (16%)	148 (11%)
	3	9,097	5,128 (56%)	1,988 (22%)	1,272 (14%)	709 (8%)
	4	9,631	3,519 (36%)	2,476 (26%)	2,203 (23%)	1,433 (15%)
24-hour	1	8,370	4,170 (50%)	1,765 (21%)	1,378 (16%)	1,057 (13%)
	2	1,025	503 (49%)	206 (20%)	155 (15%)	161 (16%)
	3	8,635	4,503 (52%)	1,527 (18%)	1,466 (17%)	1,139 (13%)
	4	9,325	3,316 (36%)	2,278 (24%)	2,171 (23%)	1,560 (17%)
96-hour	1	8,415	3,990 (47%)	1,551 (18%)	1,389 (17%)	1,485 (18%)
	2	1,134	542 (48%)	228 (20%)	188 (16%)	176 (16%)
	3	8,653	4,055 (47%)	1,720 (20%)	1,463 (17%)	1,415 (16%)
	4	8,908	3,696 (41%)	1,962 (22%)	1,653 (19%)	1,597 (18%)

Table A.1. Total number of 24-hour precipitation cases in each quartile for the North Plains climate region (Reproduced from NOAA Atlas 14 Volume 8 Version 2).

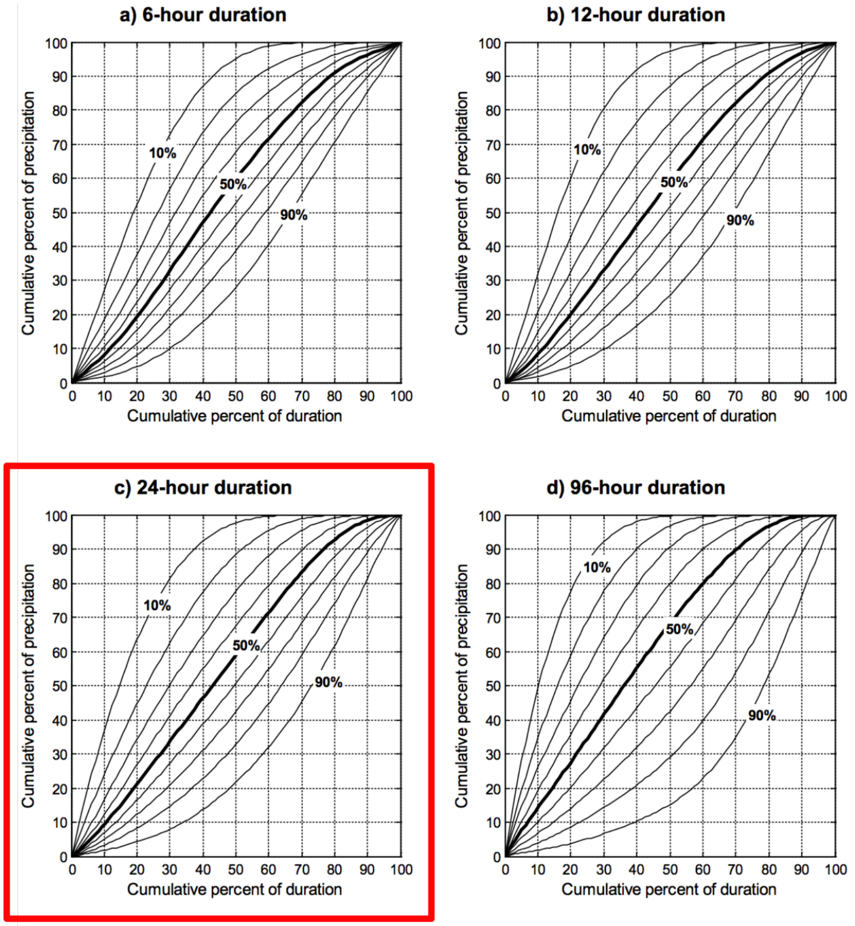
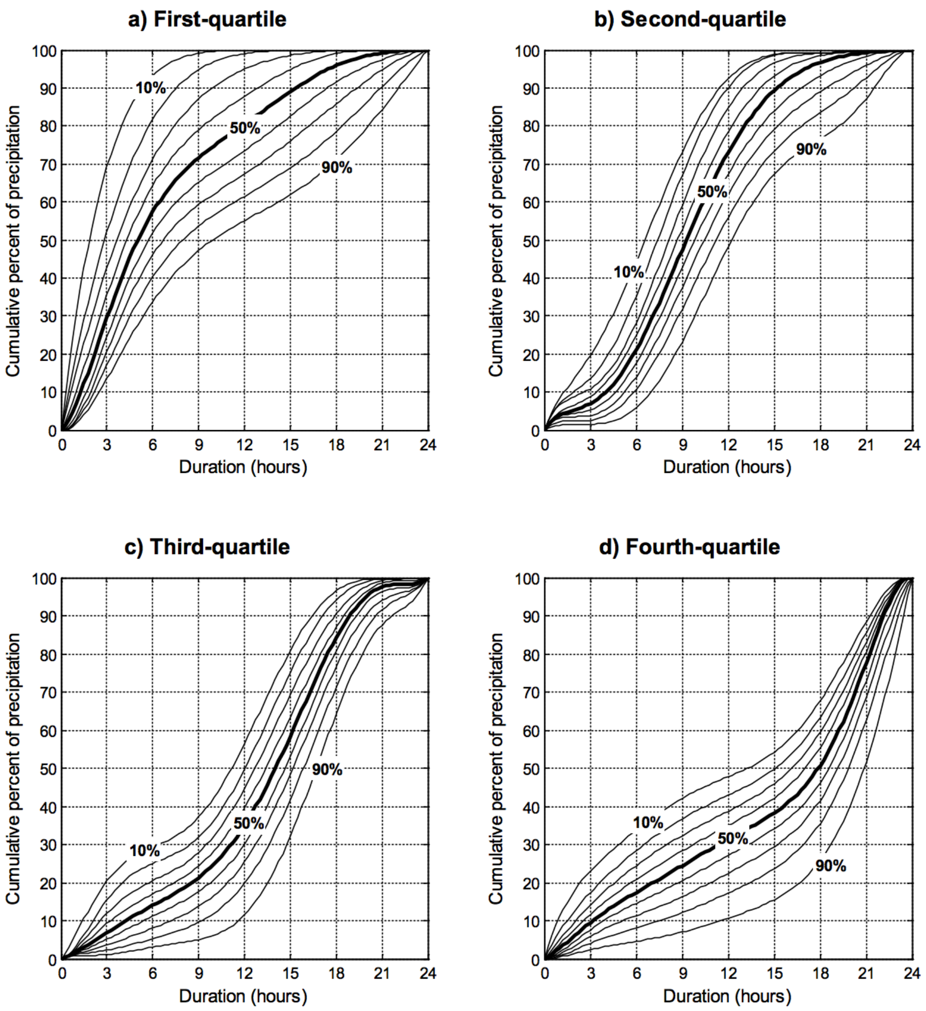


Figure A.5.1. Temporal distribution curves for the Mississippi Valley region (region 4) all cases for: a) 6-hour, b) 12-hour, c) 24-hour, and d) 96-hour durations.

Figure A.5. Example temporal distribution curves for cumulative precipitation. Curves for 24-hour duration are highlighted.



a) First-quartile

b) Second-quartile

c) Third-quartile

d) Fourth-quartile

Figure A.5.4. 24-hour temporal distribution curves for the Mississippi Valley region (region 4): a) first-quartile, b) second-quartile, c) third-quartile, and d) fourth-quartile cases.

Figure A.6. Example temporal distribution curves for cumulative precipitation. Curves for 24-hour duration are highlighted.

A.5.2 Reachability Analysis (continued)

Discrete, Time-Invariant Systems

In Chapter 6, reachability analysis for discrete time systems was used to estimate the best initial state for each storage node. This can be extended to obtain a sequence of control inputs to reach this desired state.

First, define:

$$\xi(k) = x(k) - x_0(k) - x_d(k) = x_u(k)$$

and let:

$$\hat{\xi}\,(k) = \text{Desired End State}$$

Then, a sequence of control inputs to reach that state is given by:

$$u(j) = \mathbf{B}_\mathbf{u}^T \left(\mathbf{A}^{(k-1-j)}\right)^T \mathbf{Gr}^{-1}(k)\ \xi\hat{(k)}\,,\ for\,j\ = 0,\ 1,\ 2,\ \dots$$

where

$$\mathbf{Gr}(k) = \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\,\mathbf{B_u}\mathbf{B}_\mathbf{u}^T\left(\mathbf{A}^{(k-1-j)}\right)^T$$

then

$$\hat{\xi}\,(k) = \xi(k)$$

This sequence is not unique, as all control inputs to reach $\hat{\xi}\,(k)$ is given by:

$$u(j) = \mathbf{B}_\mathbf{u}^T \left(\mathbf{A}^{(k-1-j)}\right)^T \mathbf{Gr}^{-1}(k)\ \xi\hat{(k)} + \tilde{u}\,(j),\ for\,j\ = 0,\ 1,\ 2,\ \dots$$

where $\tilde{u}$ is any sequence such that:

$$0 = \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\,\mathbf{B_u}\ \tilde{u}\,(j)$$

Discrete Time-Varying Systems

Note that this is further complicated for time-varying systems if the system is dependent on the current state as well as the current timestep. However, if these depend explicitly on time or are known a priori, such as for tracking purposes, a similar analysis can be performed

For a state-space representation of a discrete, time-varying system:

$$x(k+1) = \mathbf{A}(k)\,x(k) + \mathbf{B_u}(k)\,u(k) + \mathbf{B_d}(k)\,d(k)$$

Let $\psi(k, j)$ be the state transition matrix:

$$\psi(k,\,j) = \prod_{\ell=j}^{k-1} \mathbf{A}\,(\ell)$$

Then,

$$x(k) = x_0(k) + x_u(k) + x_d(k)$$

$$x_0(k) = \psi(k,\,0)\,x(0)$$

$$x_u(k) = \sum_{j=0}^{k-1} \psi\,(k,\,j)\mathbf{B_u}(j)\,u(j)$$

$$x_d(k) = \sum_{j=0}^{k-1} \psi\,(k,\,j)\mathbf{B_d}(k)\,d(j)$$

Now the estimated initial state $\hat{x}\,(0)$ to reach the desired final state $x(N)$ can be approximated by:

$$\hat{x}\,(0) = \psi(k,\,j)^{-1}\left(x(N) - \sum_{j=0}^{N-1} \psi\,(k,\,j)\mathbf{B_d}(j)\,d(j) \right)$$

And a sequence of control inputs to reach that state is given by:

$$u(j) = \mathbf{B}_\mathbf{u}^T\,(\psi(k,\,j))^T\,\mathbf{Gr}^{-1}(k)\ \hat{\xi(k)} + \tilde{u}\,(j),\ for\, j\ = 0,\,1,\,2,\,\ldots$$

where

$$\mathbf{Gr}(k) = \sum_{j=0}^{k-1} \psi\,(k,\,j)\,\mathbf{B_u}\mathbf{B}_\mathbf{u}^T(\psi(k,\,j))^T$$

and $\tilde{u}$ is any sequence such that:

$$0 = \sum_{j=0}^{k-1} \psi\,(k,\,j)\,\mathbf{B_u}\ \tilde{u}\,(j)$$

Bibliography

[1] A. Bronstert, D. Niehoff and G. Bürger, '"Effects of climate and land-use change on storm runoff generation: present knowledge and modelling capabilities," *Hydrol.Process.*, vol. 16, no. 2, pp. 509-529.

[2] A.D. Gronewold, V. Fortin, B. Lofgren, A. Clites, C.A. Stow and F. Quinn, '"Coasts, water levels, and climate change: A Great Lakes perspective," *Clim.Change*, vol. 120, no. 4, pp. 697-711.

[3] N. Knowles, M.D. Dettinger and D.R. Cayan, '"Trends in snowfall versus rainfall in the western United States," *J.Clim.*, vol. 19, no. 18, pp. 4545-4559.

[4] S. Doocy, A. Daniels, C. Packer, A. Dick and T.D. Kirsch, '"The human impact of floods: a historical review of events 1980-2009 and systematic literature review," *PLoS Curr.*, vol. 5, Apr 16, pp. https: //doi.org/10.1371%2Fcurrents.dis.f4deb457904936b07c09daa98ee8171a.

[5] C.J. Vörösmarty, P.B. McIntyre, M.O. Gessner, D. Dudgeon, A. Prusevich, P. Green, S. Glidden, S.E. Bunn, C.A. Sullivan and C. Reidy Liermann, '"Global threats to human water security and river biodiversity," *Nature*, vol. 467, no. 7315, pp. 555.

[6] T. Zhu, J.R. Lund, M.W. Jenkins, G.F. Marques and R.S. Ritzema, '"Climate change, urbanization, and optimal long-term floodplain protection," *Water Resour.Res.*, vol. 43, no. 6.

[7] M.A. Mallin, V.L. Johnson and S.H. Ensign, '"Comparative impacts of stormwater runoff on water quality of an urban, a suburban, and a rural stream," *Environ.Monit.Assess.*, vol. 159, no. 1, pp. 475-491.

[8] A. Semadeni-Davies, C. Hernebring, G. Svensson and L. Gustafsson, '"The impacts of climate change and urbanisation on drainage in Helsingborg, Sweden: Combined sewer system," *Journal of Hydrology*, vol. 350, no. 1, pp. 100-113.

[9] M.T. Divers, E.M. Elliott and D.J. Bain, '"Constraining nitrogen inputs to urban streams from leaking sewers using inverse modeling: implications for dissolved inorganic nitrogen (DIN) retention in urban environments," *Environ.Sci.Technol.*, vol. 47, no. 4, pp. 1816-1823.

[10] B. Sercu, Van De Werfhorst, Laurie C, J.L. Murray and P.A. Holden, '"Sewage exfiltration as a source of storm drain contamination during dry weather in urban watersheds," *Environ.Sci.Technol.*, vol. 45, no. 17, pp. 7151-7157.

[11] D.B. Booth and C.R. Jackson, '"Urbanization of aquatic systems: degradation thresholds, stormwater detection, and the limits of mitigation," *J.Am.Water Resour.Assoc.*, vol. 33, no. 5, pp. 1077-1090.

[12] J.K. Finkenbine, J. Atwater and D. Mavinic, '"Stream health after urbanization," *JAWRA Journal of the American Water Resources Association*, vol. 36, no. 5, pp. 1149-1160.

[13] L. Wang, J. Lyons, P. Kanehl and R. Bannerman, '"Impacts of urbanization on stream habitat and fish across multiple spatial scales," *Environ.Manage.*, vol. 28, no. 2, pp. 255-266.

[14] J. Barco, T.S. Hogue, V. Curto and L. Rademacher, '"Linking hydrology and stream geochemistry in urban fringe watersheds," *Journal of Hydrology*, vol. 360, no. 1, pp. 31-47.

[15] L. Sahagun, '"High cost of fighting urban runoff examined in report," *LA Times*.

[16] M. Wines, '"Behind Toledo's Water Crisis, a Long-Troubled Lake Erie," *New York Times*.

[17] S. Doughton, '"Toxic road runoff kills adult coho salmon in hours, study finds," *Seattle Times*.

[18] J.H. Ahn, S.B. Grant, C.Q. Surbeck, P.M. DiGiacomo, N.P. Nezlin and S. Jiang, '"Coastal water quality impact of stormwater runoff from an urban watershed in southern California," *Environ.Sci.Technol.*, vol. 39, no. 16, pp. 5940-5953.

[19] R.O. Carey, W.M. Wollheim, G.K. Mulukutla and M.M. Mineau, '"Characterizing storm-event nitrate fluxes in a fifth order suburbanizing watershed using in situ sensors," *Environ.Sci.Technol.*, vol. 48, no. 14, pp. 7756-7765.

[20] United States Environmental Protection Agency, '"Nonpoint Source Pollution: The Nation's Largest Water Quality Problem,", vol. EPA-841 -F-96-004A.

[21] E.A. Rosenberg, P.W. Keys, D.B. Booth, D. Hartley, J. Burkey, A.C. Steinemann and D.P. Lettenmaier, '"Precipitation extremes and the impacts of climate change on stormwater infrastructure in Washington State," *Clim.Change*, vol. 102, no. 1, pp. 319-349.

[22] R. Hawley and G. Vietz, '"Addressing the urban stream disturbance regime," *Freshwater Science*, pp. 000-000.

[23] L.S. Coffman, R. Goo and R. Frederick, '"Low-impact development: An innovative alternative approach to stormwater management,", pp. 1-10.

[24] E.W. Strecker, M.M. Quigley, B.R. Urbonas, J.E. Jones and J.K. Clary, '"Determining urban storm water BMP effectiveness," *J.Water Resour.Plann.Manage.*, vol. 127, no. 3, pp. 144-149.

[25] A. Askarizadeh, M.A. Rippy, T.D. Fletcher, D.L. Feldman, J. Peng, P. Bowler, A.S. Mehring, B.K. Winfrey, J.A. Vrugt and A. AghaKouchak, '"From rain tanks to catchments: use of low-impact development to address hydrologic symptoms of the urban stream syndrome," *Environ.Sci.Technol.*, vol. 49, no. 19, pp. 11264-11280.

[26] W.F. Hunt, A.P. Davis and R.G. Traver, '"Meeting hydrologic and water quality goals through targeted bioretention design," *J.Environ.Eng.*, vol. 138, no. 6, pp. 698-707.

[27] M. Keeley, A. Koburger, D.P. Dolowitz, D. Medearis, D. Nickel and W. Shuster, '"Perspectives on the use of green infrastructure for stormwater management in Cleveland and Milwaukee," *Environ.Manage.*, vol. 51, no. 6, pp. 1093-1108.

[28] A.E. Barbosa, J.N. Fernandes and L.M. David, '"Key issues for sustainable urban stormwater management," *Water Res.*, vol. 46, no. 20, pp. 6787-6798.

[29] G. Petrucci, E. Rioust, J. Deroubaix and B. Tassin, '"Do stormwater source control policies deliver the right hydrologic outcomes?" *Journal of Hydrology*, vol. 485, pp. 188-200.

[30] L. Atzori, A. Iera and G. Morabito, '"The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787-2805.

[31] J.T. Crawford, L.C. Loken, N.J. Casson, C. Smith, A.G. Stone and L.A. Winslow, '"High-speed limnology: Using advanced sensors to investigate spatial variability in biogeochemistry and hydrology," *Environ.Sci.Technol.*, vol. 49, no. 1, pp. 442-450.

[32] B.A. Pellerin, B.A. Stauffer, D.A. Young, D.J. Sullivan, S.B. Bricker, M.R. Walbridge, G.A. Clyde and D.M. Shaw, '"Emerging tools for continuous nutrient monitoring networks: Sensors advancing science and water resources protection," *JAWRA Journal of the American Water Resources Association*, vol. 52, no. 4, pp. 993-1008.

[33] A. Dienhart, K. Erickson, C. Wennen, M. Henjum, R. Hozalski, P. Novak and W. Arnold, '"In situ sensors for measuring pollutant loads in urban streams and evaluating stormwater BMP performance,", pp. 870-879.

[34] L. Atassi and A. Tobias, '"Regional sewer district chooses costly tunnels over 'green' infrastructure, though vacant land abounds in Cleveland,", vol. 2017, no. September 4.

[35] K. Ridgeway and M. Rabbaig, '"A Dam Good Idea." *Water Environment Federation Magazine*.

[36] Anonymous '"Open Storm Consortium,", vol. 2017, no. September 04.

[37] R. Katebi, M.A. Johnson and J. Wilkie, '"Control and instrumentation for wastewater treatment plants,", 2012.

[38] Gaborit, E and Muschalla, D and Vallet, B and others, '"Improving the performance of stormwater detention basins by real-time control using rainfall forecasts," *Urban Water J.*, vol. 10, no. 4, 1~ # aug, pp. 230-246.

[39] J.R. Middleton and M.E. Barrett, '"Water quality performance of a batch-type stormwater detention basin," *Water Environ.Res.*, vol. 80, no. 2, pp. 172-178.

[40] C. Jacopin, E. Lucas, M. Desbordes and P. Bourgogne, '"Optimisation of operational management practices for the detention basins," *Water science and technology*, vol. 44, no. 2-3, pp. 277-285.

[41] J.F. Carpenter, B. Vallet, G. Pelletier, P. Lessard and P.A. Vanrolleghem, '"Pollutant removal efficiency of a retrofitted stormwater detention pond," *Water Quality Research Journal*, vol. 49, no. 2, pp. 124-134.

[42] D. Muschalla, B. Vallet, F. Anctil, P. Lessard, G. Pelletier and P.A. Vanrolleghem, '"Ecohydraulic-driven real-time control of stormwater basins," *Journal of Hydrology*, vol. 511, pp. 82-91.

[43] A. Poresky, R. Boyle and O. Cadwalader, '"Taking Stormwater Real Time Controls to the Watershed Scale: Evaluating the Business Case and Developing an Implementation Roadmap for an Oregon MS4.".

[44] A. Gilpin and M. Barrett, '"Interim Report on the Retrofit of an Existing Flood Control Facility to Improve Pollutant Removal in an Urban Watershed,", pp. 65-74.

[45] L. Montestruque and M. Lemmon, '"Globally Coordinated Distributed Storm Water Management System,", pp. 10.

[46] C. Sammut and G.I. Webb, '"Encyclopedia of machine learning,", 2011.

[47] C.H. Emerson, C. Welty and R.G. Traver, '"Watershed-scale evaluation of a system of storm water detention basins," *J.Hydrol.Eng*.

[48] A.P. Davis, W.F. Hunt, R.G. Traver and M. Clar, '"Bioretention technology: Overview of current practice and future needs," *J.Environ.Eng*., vol. 135, no. 3, pp. 109-117.

[49] J. Hathaway, W. Hunt and S. Jadlocki, '"Indicator bacteria removal in storm-water best management practices in Charlotte, North Carolina," *J.Environ.Eng*., vol. 135, no. 12, pp. 1275-1285.

[50] C.K. Makropoulos and D. Butler, '"Distributed water infrastructure for sustainable communities," *Water Resour.Manage*., vol. 24, no. 11, pp. 2795-2816.

[51] S. Christodoulou, A. Agathokleous, A. Kounoudes and M. Mills, '"Wireless sensor networks for water loss detection," *European Water*, vol. 30, pp. 41-48.

[52] N. Jin, R. Ma, Y. Lv, X. Lou and Q. Wei, '"A novel design of water environment monitoring system based on WSN,", vol. 2, pp. V2-593-V2-597.

[53] R. Kübert, G. Katsaros and T. Wang, '"A RESTful implementation of the WS-Agreement specification,", pp. 67-72.

[54] L. Atzori, A. Iera and G. Morabito, '"The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787-2805.

[55] D. Christin, A. Reinhardt, P.S. Mogre and R. Steinmetz, '"Wireless sensor networks and the internet of things: selected challenges," *Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*, pp. 31-34.

[56] E.A. Lee and S.A. Seshia, '"Introduction to embedded systems: A cyber-physical systems approach,", 2011.

[57] Y. Zhang, M. Bocquet, V. Mallet, C. Seigneur and A. Baklanov, '"Real-time air quality forecasting, part I: History, techniques, and current status," *Atmos.Environ*., vol. 60, pp. 632-655.

[58] M. Hefeeda and M. Bagheri, '"Forest fire modeling and early detection using wireless sensor networks." *Ad Hoc & Sensor Wireless Networks*, vol. 7, no. 3-4, pp. 169-224.

[59] V. Lakshmanan, T. Smith, G. Stumpf and K. Hondl, '"The warning decision support system-integrated information," *Weather and Forecasting*, vol. 22, no. 3, pp. 596-612.

[60] J. Zhang, K. Howard, C. Langston, S. Vasiloff, B. Kaney, A. Arthur, S. Van Cooten, K. Kelleher, D. Kitzmiller and F. Ding, '"National Mosaic and Multi-Sensor QPE (NMQ) system: Description, results, and future plans," *Bull.Am.Meteorol.Soc.*, vol. 92, no. 10, pp. 1321-1338.

[61] R.M. Argent, J.-. Perraud, J.M. Rahman, R.B. Grayson and G.M. Podger, '"A new approach to water quality modelling and environmental decision support systems," *Environmental Modelling & Software*, vol. 24, no. 7, pp. 809 818.

[62] A.M. Castronova, J.L. Goodall and M.M. Elag, '"Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard," *Environmental Modelling & Software*, vol. 41, pp. 72 83.

[63] J.S. Horsburgh, D.G. Tarboton, M. Piasecki, D.R. Maidment, I. Zaslavsky, D. Valentine and T. Whitenack, '"An integrated system for publishing environmental observations data," *Environmental Modelling & Software*, vol. 24, no. 8, pp. 879 888.

[64] A.D. Gronewold, A.H. Clites, J.P. Smith and T.S. Hunter, '"A dynamic graphical interface for visualizing projected, measured, and reconstructed surface water elevations on the earth's largest lakes," *Environmental Modelling & Software*, vol. 49, pp. 34-39.

[65] J.L. Goodall, J.S. Horsburgh, T.L. Whiteaker, D.R. Maidment and I. Zaslavsky, '"A first approach to web services for the National Water Information System," *Environmental Modelling & Software*, vol. 23, no. 4, pp. 404 411.

[66] P. Taylor, D. Valentine, G. Walker, P. Sheahan, A. Pratt, I. Dornblut, S. Grellet, C. Heidmann, J. Wilhelmi and M. Christian, '"Harmonising Standards for Water Observation Data-Discussion Paper," *Open Geospatial Consortium.*

[67] S. Tilak, P. Hubbard, M. Miller and T. Fountain, '"The ring buffer network bus (RBNB) dataturbine streaming data middleware for environmental observing systems,", pp. 125-133.

[68] T. Fountain, S. Tilak, P. Hubbard, P. Shin and L. Freudinger, '"The Open Source DataTurbine Initiative: Streaming data middleware for environmental observing systems,".

[69] M. Stocker, E. Baranizadeh, H. Portin, M. Komppula, M. Rönkkö, A. Hamed, A. Virtanen, K. Lehtinen, A. Laaksonen and M. Kolehmainen, '"Representing situational knowledge acquired from sensor data for atmospheric phenomena," *Environmental Modelling & Software*, vol. 58, pp. 27-47.

[70] R. Rea, '"IBM InfoSphere Streams."Redefining Real Time Analytic Processing"," *IBM Software, Thought Leadership White Paper*, pp. 1-8.

[71] G.P. Davis and R.K. Rew, '"The Unidata LDM: programs and protocols for flexible processing of data products,".

[72] M. Daniels, V. Chandrasekar, S. Graves, S. Harper, B. Kerkez and F. Vernon, '"Executive Summary: Earthcube Real-time Workshop Results,".

[73] L. Díaz, A. Bröring, D. McInerney, G. Libertá and T. Foerster, '"Publishing sensor observations into Geospatial Information Infrastructures: A use case in fire danger assessment," *Environmental Modelling & Software*, vol. 48, pp. 65-80.

[74] C. Hartung, R. Han, C. Seielstad and S. Holbrook, '"FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments,", pp. 28-41.

[75] E. Fleisch, '"What is the internet of things? An economic perspective," *Economics, Management, and Financial Markets*, no. 2, pp. 125-157.

[76] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, '"Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Comput.Syst.*, vol. 29, no. 7, pp. 1645-1660.

[77] L. Mainetti, L. Patrono and A. Vilei, '"Evolution of wireless sensor networks towards the internet of things: A survey,", pp. 1-6.

[78] G.K. Teklemariam, J. Hoebeke, I. Moerman and P. Demeester, '"Facilitating the creation of IoT applications through conditional observations in CoAP," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1-19.

[79] P. Thubert, T. Watteyne, M.R. Palattella, X. Vilajosana and Q. Wang, '"IETF 6TSCH: combining IPv6 connectivity with industrial performance,", pp. 541-546.

[80] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser and K. Pister, '"OpenWSN: a standards-based low-power wireless development environment," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480-493.

[81] P. Lamela Seijas, H. Li and S. Thompson, '"Towards property-based testing of RESTful web services,", pp. 77-78.

[82] G. Wang, '"Improving data transmission in web applications via the translation between XML and JSON,", pp. 182-185.

[83] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, '"Context aware computing for the internet of things: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 414-454.

[84] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, '"Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Comput.Syst.*, vol. 25, no. 6, pp. 599-616.

[85] Y. Zhou, D.R. Obenour, D. Scavia, T.H. Johengen and A.M. Michalak, '"Spatial and Temporal Trends in Lake Erie Hypoxia, 1987–2007," *Environ.Sci.Technol.*, vol. 47, no. 2, pp. 899 905.

[86] D. Scavia, M.A. Evans and D.R. Obenour, '"A Scenario and Forecast Model for Gulf of Mexico Hypoxic Area and Volume," *Environ.Sci.Technol.*, pp. 130904143022002.

[87] M.R. Gartia, B. Braunschweig, T.W. Chang, P. Moinzadeh, B.S. Minsker, G. Agha, A. Wieckowski, L.L. Keefer and G.L. Liu, '"The microelectronic wireless nitrate sensor network for environmental water monitoring," *J.Environ.Monit.*, vol. 14, no. 12, Dec, pp. 3068-3075.

[88] D. Diamond, S. Coyle, S. Scarmagnani and J. Hayes, '"Wireless Sensor Networks and Chemo-/Biosensing," *Chem.Rev.*, vol. 108, no. 2, pp. 652 679.

[89] H.E. Gall, C.T. Jafvert and B. Jenkinson, '"Integrating hydrograph modeling with real-time flow monitoring to generate hydrograph-specific sampling schemes," *Journal of Hydrology*, vol. 393, no. 3, pp. 331.

[90] R. Field, '"Wet-Weather Flow in the Urban Watershed: Technology and Management,", 2002.

[91] J.J. Sansalone and C.M. Cristina, '"First Flush Concepts for Suspended and Dissolved Solids in Small Impervious Watersheds," *J.Environ.Eng.*, vol. 130, no. 11, pp. 1301 1314.

[92] K. Årzén, '"A simple event-based PID controller,", vol. 18, pp. 423-428.

[93] K.J. Åström and B. Bernhardsson, '"Comparison of periodic and event based sampling for first-order stochastic systems,", vol. 11, pp. 301-306.

[94] B.P. Wong and B. Kerkez, '"Adaptive, decentralized, and real-time sampling strategies for resource constrained hydraulic and hydrologic sensor networks,", vol. 1.

[95] Y. Zhao and B. Kerkez, '"Cellular-enabled water quality measurements," *AGU Fall Meeting*, vol. 1.

[96] M. Bostock, V. Ogievetsky and J. Heer, '"D^3 data-driven documents," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2301-2309.

[97] J.G. Rowny and J.R. Stewart, '"Characterization of nonpoint source microbial contamination in an urbanizing watershed serving as a municipal water supply," *Water Res.*, vol. 46, no. 18, pp. 6143-6153.

[98] J.C. Padowski and J.W. Jawitz, '"Water availability and vulnerability of 225 large cities in the United States," *Water Resour.Res.*, vol. 48, no. 12.

[99] J.L. Montgomery, T. Harmon, C.N. Haas, R. Hooper, N.L. Clesceri, W. Graham, W. Kaiser, A. Sanderson, B. Minsker and J. Schnoor, '"The waters network: An integrated environmental observatory network for water research," *Environ.Sci.Technol.*, vol. 41, no. 19, pp. 6642-6647.

[100] D. Hill, B. Kerkez, A. Rasekh, A. Ostfeld, B. Minsker and M.K. Banks, '"Sensing and Cyberinfrastructure for Smarter Water Management: The Promise and Challenge of Ubiquity," *J.Water Resour.Plann.Manage.*, vol. 140, no. 7, pp. 01814002.

[101] R.T. Hensley, M.J. Cohen and L.V. Korhnak, '"Hydraulic effects on nitrogen removal in a tidal spring-fed river," *Water Resour.Res.*, vol. 51, no. 3, pp. 1443-1456.

[102] M.P. Miller, A.J. Tesoriero, P.D. Capel, B.A. Pellerin, K.E. Hyer and D.A. Burns, '"Quantifying watershed-scale groundwater loading and in-stream fate of nitrate using high-frequency water quality data," *Water Resour.Res.*

[103] B.A. Pellerin, B.D. Downing, C. Kendall, R.A. Dahlgren, T.E. Kraus, J. Saraceno, R.G. Spencer and B.A. Bergamaschi, '"Assessing the sources and magnitude of diurnal nitrate variability in the San Joaquin River (California) with an in situ optical nitrate sensor and dual nitrate isotopes," *Freshwat.Biol.*, vol. 54, no. 2, pp. 376-387.

[104] M. Métadier and J. Bertrand-Krajewski, '"The use of long-term on-line turbidity measurements for the calculation of urban stormwater pollutant concentrations, loads, pollutographs and intra-event fluxes," *Water Res.*, vol. 46, no. 20, pp. 6836-6856.

[105] M.E. Barrett, '"Performance comparison of structural stormwater best management practices," *Water Environ.Res.*, vol. 77, no. 1, pp. 78-86.

[106] A. Roy-Poirier, P. Champagne and Y. Filion, '"Review of bioretention system research and design: Past, present, and future," *J.Environ.Eng.*, vol. 136, no. 9, pp. 878-889.

[107] C. Ocampo-Martinez, V. Puig, G. Cembrano and J. Quevedo, '"Application of predictive control strategies to the management of complex networks in the urban water cycle [applications of control]," *IEEE Control Systems*, vol. 33, no. 1, pp. 15-41.

[108] M. Quigley and C. Brown, '"Transforming Our Cities: High-Performance Green Infrastructure," *Water Intelligence Online*, vol. 14, pp. 9781780406732.

[109] B. Kerkez, C. Gruden, M.J. Lewis, L. Montestruque, M. Quigley, B.P. Wong, A. Bedig, R. Kertesz, T. Braun and O. Cadwalader, '"Smarter Stormwater Systems," *Environ.Sci.Technol.*

[110] R. Harmel, R. Slade and R. Haney, '"Impact of sampling techniques on measured stormwater quality data for small streams," *J.Environ.Qual.*, vol. 39, no. 5, pp. 1734-1742.

[111] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, '"Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393-422.

[112] L.M. Oliveira and J.J. Rodrigues, '"Wireless sensor networks: a survey on environmental monitoring," *Journal of communications*, vol. 6, no. 2, pp. 143-151.

[113] P. Rawat, K.D. Singh, H. Chaouchi and J.M. Bonnin, '"Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1-48.

[114] J.K. Hart and K. Martinez, '"Environmental Sensor Networks: A revolution in the earth system science?" *Earth-Sci.Rev.*, vol. 78, no. 3, pp. 177-191.

[115] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia and D. Moore, '"Environmental wireless sensor networks," *Proc IEEE*, vol. 98, no. 11, pp. 1903-1917.

[116] T.D. Fletcher and A. Deletic, '"Statistical observations of a stormwater monitoring programme: lessons for the estimation of pollutant loads," *NOVATECH 2007*, vol. 3, June 25-28, 2007, pp. 1575.

[117] J. Langeveld, H. Liefting and F. Boogaard, '"Uncertainties of stormwater characteristics and removal rates of stormwater treatment facilities: Implications for stormwater handling," *Water Res.*, vol. 46, no. 20, pp. 6868-6880.

[118] M.K. Leecaster, K. Schiff and L.L. Tiefenthaler, '"Assessment of efficient sampling designs for urban stormwater monitoring," *Water Res.*, vol. 36, no. 6, pp. 1556-1564.

[119] M.A. Benedict and E.T. McMahon, '"Green infrastructure," *Island, Washington, DC.*

[120] A. Barbosa, J. Fernandes and L. David, '"Key issues for sustainable urban stormwater management," *Water Res.*, vol. 46, no. 20, pp. 6787-6798.

[121] D.K. Rucinski, D. Beletsky, J.V. DePinto, D.J. Schwab and D. Scavia, '"A simple 1-dimensional, climate based dissolved oxygen model for the central basin of Lake Erie," *J.Great Lakes Res.*, vol. 36, no. 3, pp. 465-476.

[122] W.J. Edwards, J.D. Conroy and D.A. Culver, '"Hypolimnetic oxygen depletion dynamics in the central basin of Lake Erie," *J.Great Lakes Res.*, vol. 31, pp. 262-271.

[123] D. McCarthy, J. Hathaway, W. Hunt and A. Deletic, '"Intra-event variability of Escherichia coli and total suspended solids in urban stormwater runoff," *Water Res.*, vol. 46, no. 20, pp. 6661-6670.

[124] A. Eleria and R.M. Vogel, *Predicting Fecal Coliform Bacteria Levels in the Charles River, Massachusetts, Usa1.*

[125] J.M. Hathaway and W.F. Hunt, '"Evaluation of first flush for indicator bacteria and total suspended solids in urban stormwater runoff," *Water, Air, & Soil Pollution*, vol. 217, no. 1-4, pp. 135-147.

[126] P.M. Bach, D.T. McCarthy and A. Deletic, '"Redefining the stormwater first flush phenomenon," *Water Res.*, vol. 44, no. 8, pp. 2487-2498.

[127] J.H. Lee and K.W. Bang, '"Characterization of urban stormwater runoff," *Water Res.*, vol. 34, no. 6, pp. 1773-1780.

[128] J. Bertrand-Krajewski, G. Chebbo and A. Saget, '"Distribution of pollutant mass vs volume in stormwater discharges and the first flush phenomenon," *Water Res.*, vol. 32, no. 8, pp. 2341-2356.

[129] A. Deletic, '"The first flush load of urban surface runoff," *Water Res.*, vol. 32, no. 8, pp. 2462-2470.

[130] G.W. Characklis and M.R. Wiesner, '"Particles, metals, and water quality in runoff from large urban watershed," *J.Environ.Eng.*, vol. 123, no. 8, pp. 753-759.

[131] J.S. Horsburgh, A.S. Jones, D.K. Stevens, D.G. Tarboton and N.O. Mesner, '"A sensor network for high frequency estimation of water quality constituent fluxes using surrogates," *Environmental Modelling & Software*, vol. 25, no. 9, pp. 1031-1044.

[132] R.J. Wagner, R.W. Boulger Jr, C.J. Oblinger and B.A. Smith, '"Guidelines and standard procedures for continuous water-quality monitors: station operation, record computation, and data reporting," *U.S. Geological Survey Techniques and Methods 1-D3*, pp. 51.

[133] M. USEPA, '"Methods for chemical analysis of water and wastes," *Cincinnati, Ohio, USA.*

[134] R. Harmel, R. Cooper, R. Slade, R. Haney and J. Arnold, '"Cumulative uncertainty in measured streamflow and water quality data for small watersheds," *Transactions-American Society of Agricultural Engineers*, vol. 49, no. 3, pp. 689.

[135] R. Harmel, K. King and R. Slade, '"Automated storm water sampling on small watersheds," *Appl.Eng.Agric.*, vol. 19, no. 6, pp. 667-674.

[136] C.W. Anderson and S. Rounds, '"Use of continuous monitors and autosamplers to predict unmeasured water-quality constituents in tributaries of the Tualatin River, Oregon,", 2010.

[137] H.J. Beck and G.F. Birch, '"Metals, nutrients and total suspended solids discharged during different flow conditions in highly urbanised catchments," *Environ.Monit.Assess.*, vol. 184, no. 2, pp. 637-653.

[138] A.J. Horowitz, '"Monitoring suspended sediments and associated chemical constituents in urban environments: lessons from the city of Atlanta, Georgia, USA Water Quality Monitoring Program," *Journal of Soils and Sediments*, vol. 9, no. 4, pp. 342-363.

[139] A. Singh, R. Nowak and P. Ramanathan, '"Active learning for adaptive mobile sensing networks,", pp. 60-68.

[140] A. Krause, R. Rajagopal, A. Gupta and C. Guestrin, '"Simultaneous placement and scheduling of sensors," *2009 International Conference on Information Processing in Sensor Networks*, pp. 181.

[141] J. Reeves, J. Chen, X.L. Wang, R. Lund and Q.Q. Lu, '"A Review and Comparison of Changepoint Detection Techniques for Climate Data," *Journal of Applied Meteorology and Climatology*, vol. 46, no. 6, pp. 900 915.

[142] A. Tartakovsky, I. Nikiforov and M. Basseville, '"Sequential analysis: Hypothesis testing and changepoint detection,", 2014.

[143] T. Hastie, R. Tibshirani and J. Friedman, '"Unsupervised learning,", pp. 485-585.

[144] Weather Underground, '"Weather Underground,", vol. 2016, no. 01/12.

[145] D.B. Baker, R.P. Richards, T.T. Loftus and J.W. Kramer, '"A new flashiness index: characteristics and applications to Midwestern rivers and streams," *J.Am.Water Resour.Assoc.*, vol. 40, no. 2, pp. 503-522.

[146] B.P. Wong, Y. Zhao and B. Kerkez, '"A Web service-based architecture for real-time hydrologic sensor networks,".

[147] Teledyne Isco, '"Teledyne Isco,", vol. 2016, no. May 31.

[148] R. Grayson, B.L. Finlayson, C. Gippel and B. Hart, '"The potential of field turbidity measurements for the computation of total phosphorus and suspended solids loads," *J.Environ.Manage.*, vol. 47, no. 3, pp. 257-267.

[149] H. Rügner, M. Schwientek, B. Beckingham, B. Kuch and P. Grathwohl, '"Turbidity as a proxy for total suspended solids (TSS) and particle facilitated pollutant transport in catchments," *Environmental earth sciences*, vol. 69, no. 2, pp. 373-380.

[150] D.M. Dolan and K.P. McGunagle, '"Lake Erie total phosphorus loading analysis and update: 1996–2002," *J.Great Lakes Res.*, vol. 31, pp. 11-22.

[151] L. Li, C. Yin, Q. He and L. Kong, '"First flush of storm runoff pollution from an urban catchment in China," *Journal of Environmental Sciences*, vol. 19, no. 3, pp. 295-299.

[152] Y. Zhang, N. Meratnia and P. Havinga, '"Outlier detection techniques for wireless sensor networks: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 159-170.

[153] CUAHSI, '"CUASHI,", vol. 2016, no. 01/12.

[154] I. InfluxData, '"InfluxDB - Time-Series Data Storage | InfluxData,", vol. 2016, no. 10/01.

[155] I. Amazon Web Services, '"Amazon Web Services (AWS) - Cloud Computing Services,", vol. 2016, no. 10/01.

[156] M.E. Ellison and M.T. Brett, '"Particulate phosphorus bioavailability as a function of stream flow and land cover," *Water Res.*, vol. 40, no. 6, pp. 1258-1268.

[157] M.J. Paul and J.L. Meyer, '"Streams in the urban landscape," *Annu.Rev.Ecol.Syst.*, pp. 333-365.

[158] S. Wise, '"Green Infrastructure Rising," *Planning*, vol. 74, no. 8, pp. 14-19.

[159] I. Gnecco, C. Berretta, L. Lanza and P. La Barbera, '"Storm water pollution in the urban environment of Genoa, Italy," *Atmos.Res.*, vol. 77, no. 1, pp. 60-73.

[160] P. Rees, S. Long, R. Baker, D. Bordeau and R. Pei, '"Development of event-based pathogen monitoring strategies for watersheds,", 2006.

[161] L. Tiefenthaler, K. Schiff and M. Leecaster, '"Temporal variability patterns of stormwater concentrations in urban stormwater runoff," *Southern California coastal water research project annual report 1999-2000*, pp. 28-44.

[162] C.J. Walsh, A.H. Roy, J.W. Feminella, P.D. Cottingham, P.M. Groffman and R.P. Morgan, '"The urban stream syndrome: current knowledge and the search for a cure," *J.N.Am.Benthol.Soc.*, vol. 24, no. 3, pp. 706-723.

[163] B. Kerkez, C. Gruden, M.J. Lewis, L. Montestruque, M. Quigley, B.P. Wong, A. Bedig, R. Kertesz, T. Braun and O. Cadwalader, '"Smarter Stormwater Systems," *Environ.Sci.Technol.*

[164] R. Lawson, E. Riggs, D. Weiker and J. Doubek, '"Total Suspended Solids Reduction -- Implementation Plan for Malletts Creek: October 2011 - September 2016,", vol. 2017, no. September 01.

[165] B.P. Wong and B. Kerkez, '"Adaptive measurements of urban runoff quality," *Water Resour.Res.*, vol. 52, no. 11, 1~ # nov, pp. 8986-9000.

[166] K.B. Boyer and M.S. Kieser, '"URBAN STORMWATER MANGEMENT---AN MS4 SUCCESS STORY FOR WESTERN MICHIGAN UNIVERSITY," *Journal of Green Building*, vol. 7, no. 1, pp. 28-39.

[167] L. Cheng and A. AghaKouchak, '"Nonstationary precipitation Intensity-Duration-Frequency curves for infrastructure design in a changing climate," *Sci.Rep.*, vol. 4, 18~ # nov, pp. 7093.

[168] L.V. Alexander, X. Zhang, T.C. Peterson, J. Caesar, B. Gleason, A.M.G. Klein Tank, M. Haylock, D. Collins, B. Trewin, F. Rahimzadeh, A. Tagipour, K. Rupa Kumar, J. Revadekar, G. Griffiths, L. Vincent, D.B. Stephenson, J. Burn, E. Aguilar, M. Brunet, M. Taylor, M. New, P. Zhai, M. Rusticucci and J.L. Vazquez-Aguirre, '"Global observed changes in daily climate extremes of temperature and precipitation," *J.Geophys.Res.*, vol. 111, no. D5, 16~ # mar, pp. D05109.

[169] M.A. Pagano and D. Perry, '"Financing Infrastructure in the 21st Century City," *Public Works Management \& Policy*, vol. 13, no. 1, 1~ # jul, pp. 22-38.

[170] G.N. S., '"Infrastructure Report Card: Purpose and Results," *J Infrastruct Syst*, vol. 21, no. 4, 1~ # dec, pp. 02514001.

[171] T.D. Fletcher, G. Vietz and C.J. Walsh, '"Protection of stream ecosystems from urban stormwater runoff: the multiple benefits of an ecohydrological approach," *Prog.Phys.Geogr.*, vol. 38, no. 5, pp. 543-555.

[172] M.A. Benedict, E.T. McMahon and M. The Conservation, '"Green Infrastructure: Linking Landscapes and Communities,", 2012.

[173] K.M. Goff and R.W. Gentry, '"The Influence of Watershed and Development Characteristics on the Cumulative Impacts of Stormwater Detention Ponds," *Water Resour.Manage.*, vol. 20, no. 6, 1~ # dec, pp. 829-860.

[174] S. Eggimann, L. Mutzner, O. Wani, M.Y. Schneider, D. Spuhler, M. Moy de Vitry, P. Beutler and M. Maurer, '"The Potential of Knowing More: A Review of Data-Driven Urban Water Management," *Environ.Sci.Technol.*, vol. 51, no. 5, 7~ # mar, pp. 2538-2553.

[175] M. Bartos, B. Wong and B. Kerkez, '"Open storm: a complete framework for sensing and control of urban watersheds," *arXiv preprint arXiv:1708.05172*.

[176] J. Marsalek, '"Evolution of urban drainage: from cloaca maxima to environmental sustainability," *Acqua e Citta, I Convegno Nazionale di Idraulica Urbana, Cent.Stud.Idraul.Urbana, Santâ€™ Agnello di Sorrento, Italy*, pp. 28-30.

[177] E. Gaborit, D. Muschalla, B. Vallet, P. Vanrolleghem and F. Anctil, '"Improving the performance of stormwater detention basins by real-time control using rainfall forecasts," *Urban Water Journal*, vol. 10, no. 4, pp. 230-246.

[178] S. Bryant and B.M. Wadzuk, '"Modeling of a Real-Time Controlled Green Roof,", pp. 357-364.

[179] E. Gaborit, F. Anctil, G. Pelletier and P. Vanrolleghem, '"Exploring forecast-based management strategies for stormwater detention ponds," *Urban Water Journal*, vol. 13, no. 8, pp. 841-851.

[180] S.A. Wasimi and P.K. Kitanidis, '"Real-time forecasting and daily operation of a multireservoir system during floods by linear quadratic Gaussian control," *Water Resour.Res.*, vol. 19, no. 6, pp. 1511-1522.

[181] J.M. Lemos and L.F. Pinto, '"Distributed Linear-Quadratic Control of Serially Chained Systems: Application to a Water Delivery Canal [Applications of Control]," *IEEE Control Systems*, vol. 32, no. 6, pp. 26-38.

[182] S.M. Hashemy, M.J. Monem, J.M. Maestre and P.J.V. Overloop, '"Application of an In-Line Storage Strategy to Improve the Operational Performance of Main Irrigation Canals Using Model Predictive Control," *J.Irrig.Drain.Eng.*, vol. 139, no. 8, 1~ # aug, pp. 635-644.

[183] P. van Overloop, S. Weijs and S. Dijkstra, '"Multiple Model Predictive Control on a drainage canal system," *Control Eng.Pract.*, vol. 16, no. 5, pp. 531-540.

[184] P. van Overloop, '"Model Predictive Control on Open Water Systems,", 2006.

[185] X." Zhen, S.L. Yu and J. Lin, '"Optimal location and sizing of stormwater basins at watershed scale," *J.Water Resour.Plann.Manage.*, vol. 130, no. 4, pp. 339-347.

[186] J.G. Lee, A. Selvakumar, K. Alvi, J. Riverson, J.X. Zhen, L. Shoemaker and F. Lai, '"A watershed-scale design optimization model for stormwater best management practices," *Environmental Modelling & Software*, vol. 37, pp. 6-18.

[187] R. Kessler, '"Stormwater strategies: cities prepare aging infrastructure for climate change," *Environ.Health Perspect.*, vol. 119, no. 12, Dec, pp. A514-9.

[188] N.S. Grigg, '"Water, wastewater, and stormwater infrastructure management,", 2012.

[189] J. Giron\'as, L.A. Roesner, L.A. Rossman and J. Davis, '"A new applications manual for the Storm Water Management Model (SWMM)," *Environmental Modelling \& Software*, vol. 25, no. 6, 1~ # jun, pp. 813-814.

[190] L.A. Rossman, '"Storm water management model user's manual, version 5.0,", 2010.

[191] J. Schuurmans, O. Bosgra and R. Brouwer, '"Open-channel flow model approximation for controller design," *Appl.Math.Model.*, vol. 19, no. 9, pp. 525-530.

[192] X. Litrico and V. Fromion, '"Modeling and control of hydrosystems,", 2009.

[193] M. Xu, P. Van Overloop and N. Van De Giesen, '"On the study of control effectiveness and computational efficiency of reduced Saint-Venant model in model predictive control of open channel flow," *Adv.Water Resour.*, vol. 34, no. 2, pp. 282-290.

[194] M.A. Gill, '"Flood routing by the Muskingum method," *Journal of hydrology*, vol. 36, no. 3-4, pp. 353-363.

[195] J. Schuurmans, '"Control of water levels in open-channels,".

[196] P.O. Malaterre and J.P. Baume, '"Modeling and regulation of irrigation canals: existing applications and ongoing researches,", vol. 4, oct, pp. 3850-3855 o.4.

[197] A. Sinha, '"Linear systems: optimal and robust control,", 2007.

[198] F. Lin, '"Robust control design: an optimal control approach,", vol. 18, 2007.

[199] M. Dahleh, M.A. Dahleh and G. Verghese, '"Lectures on dynamic systems and control," *A �A*, vol. 4, no. 100, pp. 1-100.

[200] G. Riaño-Briceño, J. Barreiro-Gomez, A. Ramirez-Jaime, N. Quijano and C. Ocampo-Martinez, '"MatSWMM–An open-source toolbox for designing real-time control of urban drainage systems," *Environmental Modelling & Software*, vol. 83, pp. 143-154.

[201] W.J. Rawls, D.L. Brakensiek and N. Miller, '"Green-Ampt infiltration parameters from soils data," *J.Hydraul.Eng.*, vol. 109, no. 1, pp. 62-70.

[202] B.P. Wong and B. Kerkez, '"Wong Brandon Preclaro and Kerkez Branko. GitHub Repository,", vol. 2016, no. 01/12.

[203] Y. Guo, '"Hydrologic design of urban flood control detention ponds," *J.Hydrol.Eng.*

[204] U.S. Scs, '"Urban Hydrology for Small Watersheds, Technical Release No. 55 (TR-55)," *US Department of Agriculture, US Government Printing Office, Washington, DC.*

[205] K. Akitsu, C.H. Evan and S. David, '"Comparison of Synthetic Design Storms with Observed Storms in Southern Arizona," *J.Hydrol.Eng.*, vol. 16, no. 11, 1~ # nov, pp. 935-941.

[206] G.M. Bonnin, D. Martin, B. Lin, T. Parzybok, M. Yekta and D. Riley, '"Precipitation-frequency atlas of the United States," *NOAA atlas*, vol. 14, no. 2.

[207] B.P. Bledsoe, '"Stream erosion potential and stormwater management strategies," *J.Water Resour.Plann.Manage.*, vol. 128, no. 6, pp. 451-455.

[208] C. Pomeroy, N. Postel, P. O'Neill and L. Roesner, '"Development of storm-water management design criteria to maintain geomorphic stability in Kansas City metropolitan area streams," *J.Irrig.Drain.Eng.*, vol. 134, no. 5, pp. 562-566.

[209] E. Tillinghast, W. Hunt and G. Jennings, '"Stormwater control measure (SCM) design standards to limit stream erosion for Piedmont North Carolina," *Journal of Hydrology*, vol. 411, no. 3, pp. 185-196.

[210] C. MacRae, '"Experience from morphological research on Canadian streams: Is control of the two-year frequency runoff event the best basis for stream channel protection?", pp. 144-162.

[211] S.M. Nehrke and L.A. Roesner, '"Effects of design practice for flood control and best management practices on the flow-frequency curve," *J.Water Resour.Plann.Manage.*, vol. 130, no. 2, pp. 131-139.

[212] C. Rohrer and L. Roesner, '"Matching the critical portion of the flow duration curve to minimise changes in modelled excess shear," *Water science and technology*, vol. 54, no. 6-7, pp. 347-354.

[213] E.N. Pratt, '"Rules and Guidelines - Procedures and Design Criteria for Stormwater,", October 17, 2016.

[214] Y. Liu, J. Slotine and A. Barab\'asi, '"Controllability of complex networks," *Nature*, vol. 473, no. 7346, 12~ # may, pp. 167-173.

[215] R. Faragher, '"Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]," *IEEE Signal Process.Mag.*, vol. 29, no. 5, pp. 128-132.

[216] City of Los Angeles Department of Public Works, '"Stormwater Pollution Fast Facts,", vol. 2017.

[217] G. Guan, A.M. Asce, A.J. Clemmens, M. Asce, T.F. Kacerek, B.T. Wahlin and M. Asce, '"Applying Water-Level Difference Control to Central Arizona Project," *J.Irrig.Drain.Eng.*, vol. 137, no. 12, pp. 747-753.

[218] A.J. Clemmens and M. Asce, '"Water-Level Difference Controller for Main Canals," *J.Irrig.Drain.Eng.*, vol. 138, no. 1, pp. 1-8.

[219] E. Bautista, A.M. Asce, A.J. Clemmens and M. Asce, '"Volume Compensation Method for Routing Irrigation Canal Demand Changes," *J.Irrig.Drain.Eng.*, vol. 131, no. 6, pp. 494-503.

[220] E. Bautista, T.S. Strelkoff and A.J. Clemmens, '"General characteristics of solutions to the open-channel flow, feedforward control problem," *J.Irrig.Drain.Eng.*, vol. 129, no. 2, pp. 129-137.

[221] S.M. Papalexiou, D. Koutsoyiannis and C. Makropoulos, '"How extreme is extreme? An assessment of daily rainfall distribution tails," *Hydrol.Earth Syst.Sci.*, vol. 17, no. 2, 28~ # feb, pp. 851-862.

[222] H. Hanum, A.H. Wigena, A. Djuraidah and I.W. Mangku, '"Modeling extreme rainfall with Gamma-Pareto distribution," *Appl.Math.Sci.*, vol. 9, no. 121, pp. 6029-6039.

[223] H. Aksoy, '"Use of gamma distribution in hydrological analysis," *Turk.J.Eng.Environ.Sci.*, vol. 24, no. 6, pp. 419-428.

[224] G.E. Palhegyi, '"Designing storm-water controls to promote sustainable ecosystems: science and application," *J.Hydrol.Eng.*, vol. 15, no. 6, pp. 504-511.

[225] M. Morari and J. H. Lee, '"Model predictive control: past, present and future," *Comput.Chem.Eng.*, vol. 23, no. 4, 1~ # may, pp. 667-682.

[226] E.F. Camacho and C. Bordons, '"Model Predictive Control in the Process Industry,", 2012.

[227] Van Overloop, P J and Negenborn, R R and De Schutter, B and Van De Giesen, N C and Others, '"Predictive control for national water flow optimization in The Netherlands," *Intelligent infrastructures*, vol. 42, pp. 439-461.

[228] A. Zafra-Cabeza, J.M. Maestre, M.A. Ridao, E.F. Camacho and L. S\'anchez, '"Hierarchical distributed model predictive control for risk mitigation: An irrigation canal case study,", jun, pp. 3172-3177.

[229] L. Wang, '"Model Predictive Control System Design and Implementation Using MATLAB\textregistered,", 2009.

[230] J. Lipor, B.P. Wong, D. Scavia, B. Kerkez and L. Balzano, '"Distance-Penalized Active Learning Using Quantile Search," *IEEE Transactions on Signal Processing*.