INTERIM TECHNICAL REPORT ON CSONET DISTRIBUTED CONTROL ALGORITHM

M.D. LEMMON, P. WAN, AND L. LI

Abstract. CSOnet is a sensor-actuator network used to monitor and manage combined sewer overflows in municipal sewer systems. A metropolitan version of CSOnet is being built by EmNET LLC to control wastewater flows in the City of South Bend' interceptor sewer. An important part of CSOnet is the control algorithm used to manage the wastewater flows. The control algorithm was designed to maximize the diverted flow into the interceptor sewer subject to safety (no flooding) and admissibility (diverted flow is not greater than the storm inflow). The control is *decentralized* in the sense that each CSO diversion structure makes its own control decisions using the current water level in its own manhole. This technical report documents the design of the controller, the procedures used to identify plant models, the implementation of the controller in a SWMM validated model of the South Bend interceptor sewer, and the results of simulation experiments with the controller.

1. Introduction

Wireless sensor actuator networks or WSAN's consist of computer controlled sensors and actuators that communicate over a wireless (usually RF) communication network. WSAN's use sensed data to power actuators that effect the sensed environment. The resulting changes in that environment are then sensed by the network. This forms a distributed feedback loop that has the potential of enabling efficient management of geographically distributed processes.

This report describes the control algorithms that were developed for a metropolitan scale (city wide) WSAN that is currently being built by a unique partnership of private (EmNET LLC), public (City of South Bend), and academic (University of Notre Dame and Purdue University) agencies. The WSAN is being built to control the frequency of of combined-sewer overflow (CSO) events in a mid-sized U.S. city (South Bend Indiana). The system is called *CSOnet*. The problem addressed by CSO net is a major public health and environmental issue faced by many U.S. cities. South Bend's CSOnet will be completed in summer 2009 using a distributed control strategy whose development and validation are described in this report.

Date: Summer 2008.

The authors are with the University of Notre Dame's department of electrical engineering. The authors gratefully acknowledge the partial financial support of Indiana's 21st Century Technology Research Fund and the National Science Foundation (CNS-07-20457).

2. Summary of Report Findings and Accomplishments

This section summarizes the major findings and accomplishments obtained over the project's reporting period (February 2007 - July 2008). These findings/accomplishments are itemized below.

(1) Derived an optimal supervisory control strategy based on flow-rate measurements.
(2) Developed a distributed communication protocol for implementing the "optimal" supervisory control strategy.
(3) Built a hardware scale-model of an interceptor sewer testbed for EmNET LLC. Used this testbed to validate system identification methods for nodes within the system.
(4) Identified local dynamical models for each node in the South Bend interceptor sewer system.
(5) Developed a design methodology for synthesizing output feedback tracking controllers in CSOnet
(6) Designed local controllers for all nodes within the South Bend interceptor sewer system. Validated these controllers' performance against storm profiles provided by EmNET LLC.
(7) Published two conference papers on the CSOnet controller (CDC07 and MODUS08). A journal version of the MODUS 08 paper is under preparation (by invitation).
(8) Gave invited presentations of the CSOnet system to
    - Carnegie-Mellon University CenSCIR symposium
    - ARTISTE workshop on cyberphysical systems (UIUC)
    - Indiana RF Radio Alliance,
    - European Union's WIDE Project (Siena Italy, Sept 2008)
(9) Incorporated work on CSOnet in recent NSF funded project on Self-triggered networked control systems (NSF-07-20457)

The remaining tasks to be completed over next performance period

(1) Work with EmNET LLC to integrate the proposed control algorithms into the hardware testbed.
(2) Work with EmNET LLC to modify their current communication algorithms to support the CSOnet controllers.

Estimated completion time for these remaining tasks is about 1 year.

The individuals supported by this project for the reporting period include Dr. Lemmon, Mr. Pu Wan, and Ms. Lichun Li.

3. Combined-Sewer Overflow Problem

Nearly all U.S. cities in the Northeast and Midwest have sewer systems that combine sanitary and storm water flows in the same system. During rain storms, wastewater flows can easily overload these combined sewer systems, thereby forcing operators to dump the excess water into the nearest river or stream. The discharge is called a *combined sewer overflow* (CSO) event [1]. The discharged water is highly impacted with biological and chemical contaminants, thereby creating a major environmental and public health hazard. Under the provisions of the 1972 clean water act, the environmental protection agency (EPA) has begun fining municipalities for CSO events. These fines are substantial, sometimes running into the tens of millions of dollars. Municipalities have therefore begun looking for cost effective ways of reducing the frequency of CSO events.

The straightforward solution to the CSO problem is to enhance the existing sewer infrastructure by separating storm and santiary flows. Other solutions involve increasing the capacity of the wastewater treatment plant (WWTP) or building large off-line storage reservoirs. All of these options are extremely expensive and highly disruptive to the community.

Another solution uses the excess storage capacity in a city's sewer to temporarily store water during a storm. This option is referred to as *in-line storage*. The economical and reliable control of CSO events through in-line storage requires real-time monitoring and control.

Current approaches to real-time monitoring and control of sewer systems do not scale well. Sensor data is usually collected by a single computer over a Supervisory Control and Data Acquisition (SCADA) network. This computer determines the control action and distributes it back to the system through the SCADA network. It takes time to gather all of the sensor data and the delay introduced by gathering this data will also limit the rate at which control commands can be fed back to the system. Due to these delays, the control must be computed using complex simulation models of the sewer system. The entire control problem is therefore viewed as a large-scale nonlinear optimal control problem [2] which can be addressed using linear quadratic approaches [3] or model predictive control methods [4] [5]. These controllers are always implemented in a centralized fashion for very high-order plants. The system model is highly nonlinear with a great amount of uncertainty. As a result, centralized control of sewer systems tends to be complex, computationally intensive, and certainly is not robust to modeling error. All of these factors conspire to limit the scalability of centralized approaches to sewer flow control.

An alternative "decentralized" approach to CSO control was presented by Ruggaber et al [6]. This case study used an embedded network of microprocessor controlled sensors and actuators to control CSO events. The network used a simple local feedback scheme to control a stretch of sewer system fed by a 1500 foot wide by 3.2 mile long corridor. In its first month of service the network prevented a 2 million gallon CSO event. The cost of the deployed network was around $25,000, which was half of what it would have cost using existing SCADA network technologies.

The sensor-actuator network used by Ruggaber et al, therefore appeared to provide a cost-effective solution for controlling CSO events. The control used in that study was a simple switching law. It was not readily apparent how the simple controller used in [6] would scale up. A more sophisticated decentralized control strategy was developed by Wan et al. [7]. This decentralized control strategy was implemented on a simulated model of the WSAN used by Ruggaber et al. Initial results indicated that this decentralized control algorithm could reduce CSO overflows by as much as 20% over the existing passive strategies in use.

The decentralized controller of Wan et al. provided a possible way of scaling up the CSOnet controller to a metropolitan-sized network. The original algorithm, however, made a number of simplifying assumptions that could not be justified in real life. In particular, the controller proposed by Wan et al. assumed that it was possible to measure the flow rates in the pipe conduit. In general, it can be very difficult to get a consistent measurement of the turbulent flows in the sewer conduits. In addition to this, the Wan controller would sometimes result in localized flooding if a control node lacked sufficient authority to control the local flow rates. It therefore became necessary to develop a decentralized controller that could achieve its objectives using only the water level in the conduits as the primary feedback quantity. This report details the development of the required controller.

4. CSOnet Architecture

CSOnet's architecture was designed to be a set of local WSAN's that connect to an existing wide area network (WAN) through gateway devices. CSOnet can therefore be viewed as a heterogeneous sensor-actuator network. It consists of four types of devices:

- Instrumentation Node or INode: these nodes are responsible for retrieving the measurement of a given environmental variable, processing that data and forwarding the data to the destination gateway through a radio transceiver.
- Relay Node or RNode: these nodes aid in forwarding data collected by INodes that are more than one-hop away from the gateway node. The RNodes only serve to enhance the connectivity in the wireless network.
- Gateway Node or GNode: these nodes serve as gateways between the WSAN used to gather data from the INodes and a Wide Area Network (WAN) which allows remote users easy access to CSOnet's data.
- Actuator Node or ANode: these nodes are connected to valves (actuators) that are used to hold back water in the sewer system.

Figure 1 shows the prototype CSOnet built by Ruggaber et al [6]. This network shows a single ANode (marked by the "V") that receives its feedback sensor signals from three INodes (marked by the "I"). One of the INode's is located at the river to monitor actual CSO discharge into the river. The other INode's are used to measure the water level in a retention basin. The distance between the CSO outfall at the river and the retention basin is about 3 miles. Feedback information from the CSO outfall is forwarded over a a line of RNodes (marked by the "R"). This figure can be taken as a very simple example of a single WSAN. This particular WSAN has been in continuous operation since 2005 and has been extremely useful in refining CSOnet's hardware and middleware components to ensure long-life and economical maintenance.

In order to scale CSOnet up to an entire metropolitan area, it was necessary to adopt a hierarchical architecture consisting of several WSAN's that are interconnected over an existing wide area network. One reason for this is the well-known inability of WSAN's to provide an acceptable quality of service when the network becomes too large. This is a consequence of the well-known theoretical limitations on wireless network throughput [8]. Empirical studies from DARPA's NEST program [9] have suggested that flat WSANs should be limited to a diameter of 5-6 hops to prevent excessive congestion. If South Bend's

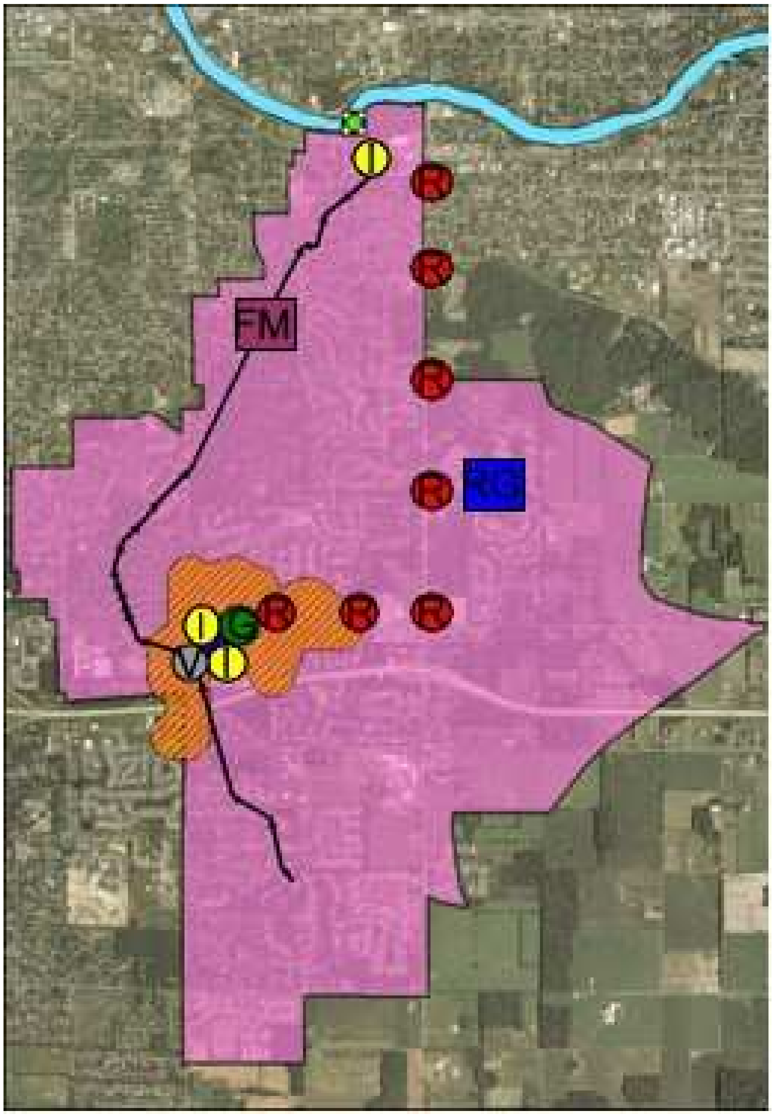


FIGURE 1. Layout of Ruggaber's [6] prototype CSOnet at Ireland-Miami Road in South Bend Indiana (2005)

CSOnet were to be built as a single flat WSAN covering the entire city, it would consist of several hundred nodes covering a 13,000 acre area. Such a deployment would require a prohibitive number of INodes and RNodes. For this reason, CSOnet consists of a set rather small WSAN's that forward their data to GNodes. The GNodes then forward the received packets to other WSAN's in the system.

To understand CSOnet's hierarchical structure, we first need to examine the actual sewer system to be controlled. Figure 2 shows a sewer system in which combined sewer trunk lines feed into a large *interceptor sewer*. Prior to 1972, municipal combined sewer lines were allowed to dump directly into rivers and streams. Under the clean water act, cities were forced to treat the water from these combined sewer lines before they were released into a river or stream. One common way to meet this regulatory burden was to build an interceptor sewer along the river. This sewer would intercept the flow from the combined sewer trunk lines and convey that flow to a wastewater treatment plant (WWTP). Under dry weather conditions the flows were small enough to be handled by the WWTP. Under wet weather conditions (storms), the flows often overwhelmed the WWTP's capacity, thereby forcing operators to dump the excess directly into the water. As noted above such discharges constitute the CSO events described earlier.

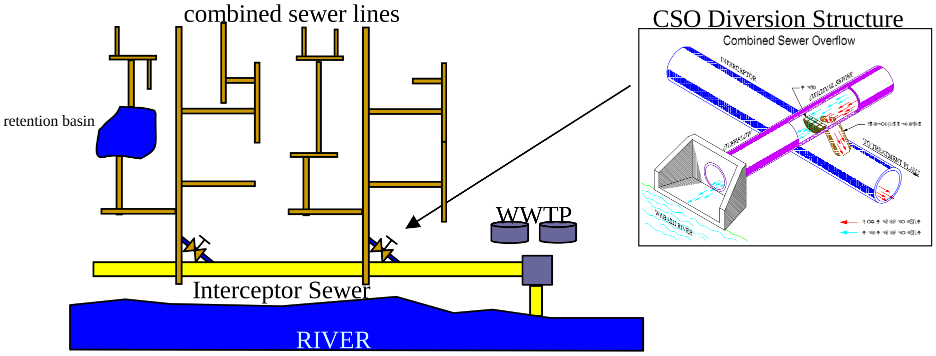


FIGURE 2. South Bend Interceptor Sewer and CSO Diversion Structure

From figure 2 we can see that the combined sewer trunk lines and interceptor sewer connect at a *CSO diversion structure*. This is the point where we can apply control. The current system in South Bend Indiana uses a passive thresholding control. When the depth of the flow is below a fixed preset threshold, the flow is diverted into the interceptor sewer. Above this threshold, the flow is dumped out into the river. The problem is that these thresholds are set for the worst-case storm scenario. By placing a WSAN in the combined sewer trunk line above the CSO diversion structure, we can estimate the actual flows into the interceptor line and thereby make closed-loop control decisions that optimize the flow into the interceptor line such that WWTP capacity limits are never exceeded. This means that the natural place to place ANodes is at the CSO diversion points. These ANodes would then adjust the amount of water diverted into the interceptor sewer line based on an adaptive threshold that is a function of the current flows into the system. Because this scheme is adaptive, it need not be as conservative as the original passive thresholding scheme.

The scenario outlined above therefore indicates that CSOnet consists of a collection of WSAN's that forward flow measurements in a combined sewer trunk line to its associated CSO diversion structure. At this diversion structure would be a GNode and ANode. The ANode would adjust the flow into the interceptor line and the GNode would serve as a gateway between this particular WSAN and neighboring WSAN's up and down the interceptor line. Figure 3 illustrates this system architecture with 2 different WSAN's controlling the two diversion structures into the interceptor line. GNodes at these diversion structures and the WWTP are used to exchange control information in a way that allows coordinated flow control across the city's entire sewer system.

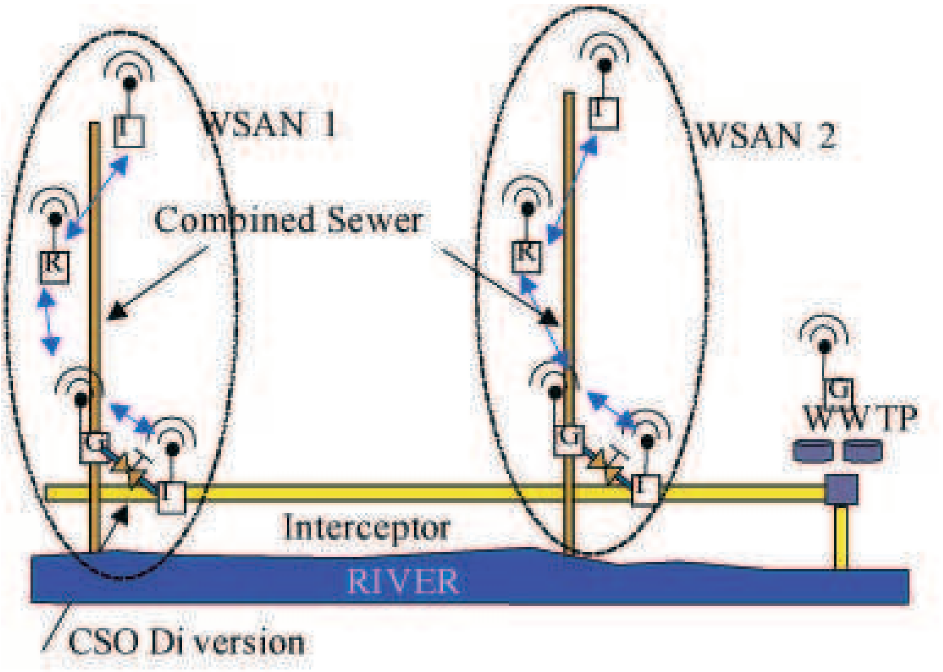


FIGURE 3. CSOnet's Hierarchical Architecture

5. OPEN CHANNEL FLOWS

The movement of water in closed unpressurized conduits can be modeled as an *open channel flow*. This section presents the fundamental equations that govern open channel flows. Simplifications are then made on those governing equations to obtain a simplified model for the plant being controlled.

The flow of water within the sewer network is modeled as an open channel flow. The most frequently used model for unsteady and non-uniform open channel flow is the *complete dynamic wave model* [10]. This model consists of two equations: the continuity equation and the momentum equation. The *continuity equation* models the conservation of mass and is given as:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1}$$

where x is the spatial coordinate along the length of the pipe (m), t is the time (s), Q is the flow (m^3/s), and A is the cross sectional area of the flow (m^2) (the water surface area perpendicular to the direction of the flow). Equation 1 says the mass of water is conserved along any closed contour in the $x - t$ plane. The *momentum equation* is

$$0 = gA\frac{\partial H}{\partial x} + \frac{\partial Q}{\partial t} - \sigma\left(2V\frac{\partial A}{\partial t} + V^2\frac{\partial A}{\partial x}\right) + gAS_f \tag{2}$$

where H is the water head level (height of water surface above the ground level)(m), $V = Q/A$ is the flow velocity (m/s), and g is the gravitational acceleration constant. The constant σ is a real number depending on the depth of the flow. S_f is called the *friction slope* which is detemrined from *Manning's equation*

$$S_f = \frac{(n/1.49)^2}{AR^{4/3}} Q|V| \tag{3}$$

where n is a constant dependent on the roughness of the conduit and $R = A/P$ where P is the "wetted" perimeter of the pipe.

Equations 1-3 form a set of nonlinear partial differential equations (PDEs) that we call the complete dynamic wave model. It is difficult to derive controllers for the systems characterized by PDEs, so we seek a simplified model which can be characterized by a set of nonlinear ordinary differential equations (ODEs). In particular, we adopt a steady (time invariant) and non-uniform (spatially variant) version of the momentum equation and an unsteady (time-varying) version of the continuity equation.

Let's consider a pipe of length L connecting two manholes as shown in figure 4. Assume that the flow rate, Q, is slowly varying. We can therefore neglect the partial derviatives, $\partial Q/\partial t$ and $\partial A/\partial t$, in equation 2 to obtain

$$\tfrac{\partial H}{\partial x} = \tfrac{H_d - H_u}{L}, \quad \tfrac{\partial A}{\partial x} = \tfrac{A_d - A_u}{L}$$

where L is the length of the pipe, H_u and H_d are the water head levels at the upstream and downstream manholes, respectively (see figure 4). A_u and A_d are the cross sectional flow areas upstream an downstream of the pipe. With these simplifications, the momentum equation can be reduced to an algebraic equation

$$H_u - H_d = kQ^2 \tag{4}$$

where k is

$$k = \frac{L(n/1.49)^2}{A^2 R^{4/3}} - \sigma(A_d - A_u)\frac{1}{gA^3} \tag{5}$$

where A is the average cross sectional flow area in the pipe. Bear in mind that k is not a constant, it is actually a function of head H_u and H_d since A_r, A_d, A_u, σ are all functions of H_u and H_d.

If we focus our attention on a single manhole (take the upstream manhole for example), then the continuity equation 1 can be rewritten as,

$$a_u \frac{dH_u}{dt} = \sum_{\text{in}} Q_{\text{in}} - \sum_{\text{out}} Q_{\text{out}} \tag{6}$$

where Q_{in} and Q_{out} are the flows that go into and out of the manhole, respectively. a_u is the water surface area of the manhole. Equation 6 simply says that the difference between the inflows and outflows for a particular manhole is equal to the rate of change in water storage.

The water network shown on top of figure 4 can be mapped to the graph shown at the bottom of figure 4. In this graph, the state of each node is given by the head level H at the manhole and the state of each link is given by the flow Q in the pipe. In this figure, the continuity equation for the upstream node takes the form

$$a_u \frac{dH_u}{dt} = Q_u - Q + w_u$$

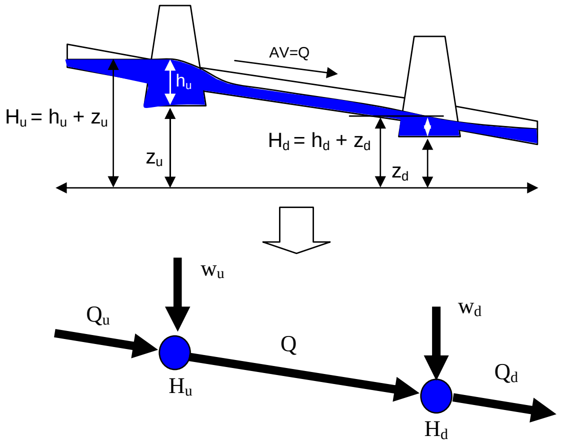


Figure 4. Simplified Model for a Single Pipe

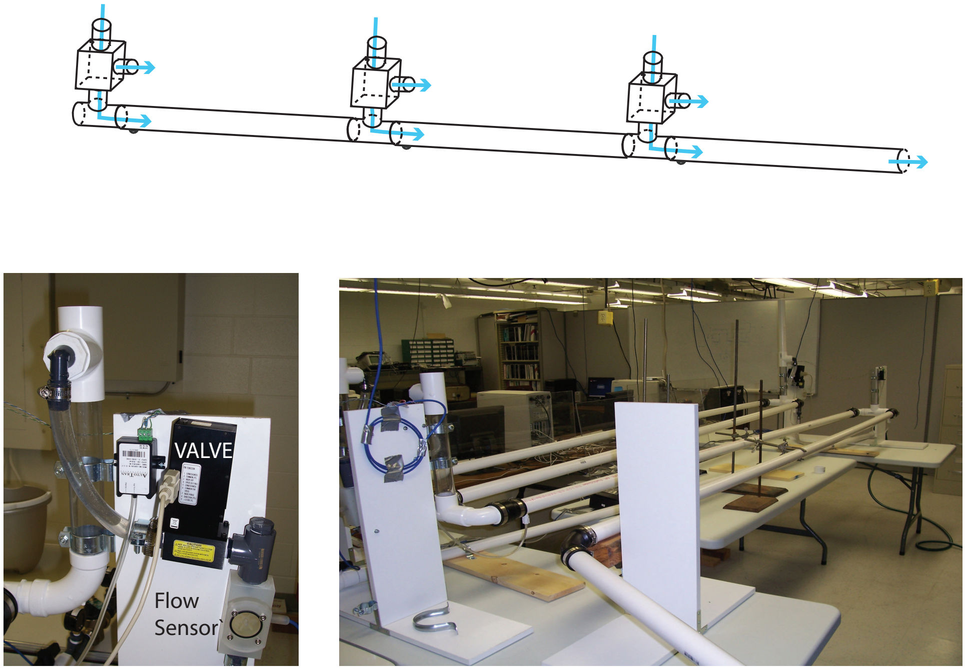
where w_u is the inflow from the storm. Throughout this paper we'll refer to pipes as links and manholes as nodes. Our simplified model therefore consists of the ODEs and algebraic equations given in equations 4-6. This simplified model was validated against a high fidelity SWMM [11] model of the interceptor sewer system in South Bend Indiana.

An important feature of the model is that the state of a node is only dependent on the state of its adjacent links, and the state of a link is only dependent on the two nodes connecting it. This highly decentralized model facilitates the design of a controller with easy decentralized implementation. Another characteristic is that when using our model the water network is very similar to an electrical network. The head H and flow Q correspond to the voltage V and current I, respectively. Equation 4 is similar to a nonlinear Ohm's law, with the "resistance" given by equation 5. Equation 6 is similar to Kirchoff's current law.

6. Experimental Model Identification

The sewer system model in equation 4-6 represents a simplification of the system that was used by Pu et al. to develop an *optimal* flow control strategy for the interceptor sewer. While this model was suitable for developing the control strategy, it turned out to be too simplistic to develop an actual feedback controller for each node in the interceptor sewer. Experiments with a hardware scale model of a 4 node interceptor sewer line showed that higher order dynamical models would be needed to control head levels on the basis of pressure measurements alone. This section reviews the hardware scaled model and its use in identifying dynamical models for water level in the interceptor line.

A scaled hardware model of four nodes in an interceptor line was built in the summer of 2007. The bottom righthand picture shows the testbed that was constructed at Notre Dame. The system consisted of 4 lengths of 2.5 inch PVC pipe. Each length of pipe was between 8 and 12 feet long. As shown in the picture, the pipes were arranged in a zig-zag manner to conserve space. Each length of pipe was supported by a wooden stand that also contained the actuation subsystem used to control how much water was injected into the system. The bottom lefthand picture shows this wooden stand containing an actuated valve and flow sensor that were controlled by a Chasqui node to limit the water inflow rate to a commanded level. The top drawing in the figure shows a straightened out version of the testbed in which each actuation subsystem is shown by a box. The water is injected from the top with an overflow out of the system and an inflow into the interceptor line.



closeup of testbed flow actuator

picture of scaled testbed model of interceptor sewer

Figure 5. Scaled Hardware Testbed of Interceptor Sewer

The scaled hardware testbed was built to help verify the simplified flow dynamics described in equations 4-6. These equations appear to be sufficiently accurate to predict steady state flows on the time scale of 10's of minutes. The predictions made by these equations using the supervisory control scheme described in [7] were validated against a high fidelity SWMM simulation model of the South Bend Interceptor sewer. These controls, however, assume direct access to both the local head level and flow rates at a specified manhole in the system. While it is possible to accurately measure head level through pressure sensors, the measurement of flow rates proves to be more expensive and noisy due to flow turbulence. It was therefore desirable to design decentralized controllers for the interceptor sewer that only used pressure measurements to control flow. For this type of controller, it was felt that we would need an input-output model of the flows through a manhole that was accurate over a wider range of time scales than was possible using the simplified flow model.

Experiments showed that the behavior of the testbed qualitatively resembles what we'd expect from the more accurate SWMM model. Figure 6 shows the head levels qualitatively match the levels predicted by a SWMM model of the 4-node testbed Node 1 matches well. Node 4 does not match, but this is due to the assumption in the model that the last node essentially has an unconstrained outfall rate. Nodes 2 and 3 match the SWMM prediction in a qualitative manner, but the match is poor enough to suggest that we need to consider a higher-order model of the node dynamics.

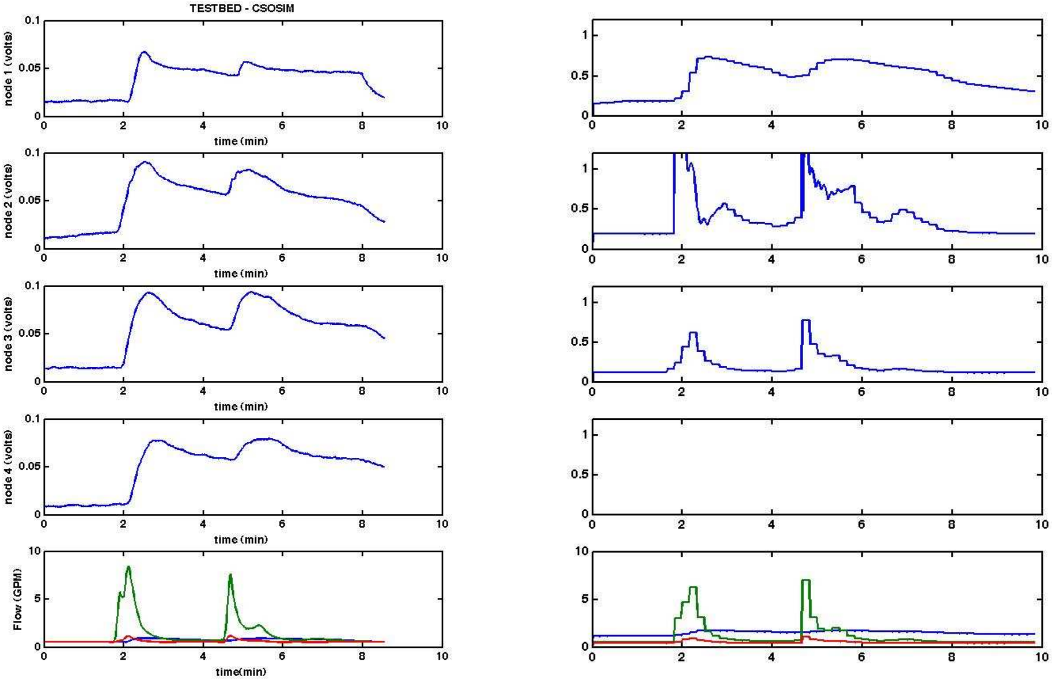


Figure 6. SWMM head (pressure) history (right) and testbed head (pressure) history (left)

This conjecture was confirmed by additional experiments that plotted a node's transient response to a known input. Figure 7 plots the pressure measured at each of the 4 nodes in the testbed as a function of the applied storm inflow. In this figure, the storm inflow was injected into node 3 (u3). The x-axis in the 4 plots is the inflow and the y-axis is the pressure. The input was a raised sinusoid that started at zero, went up to 6 gpm and then returned to zero. If the simplified model from equations 4-6 were valid, we'd expect the pressure profile for node 3 (p3 versus u3) to trace out a quadratic curve. This is not the case as can be seen in figure 7 . What we actually see is an elliptical curve which is more suggestive of a second order system. Moreover, we see that even though the input was only at node 3, there is some response at node 2. This thereby indicates that some water has been stored in the upstream link. We are therefore led to believe that the "pressure" dynamics of a node are more accurately predicted by a 2-3 order dynamical model.

Clearly the momentum equation 2 and continuity equation 1 suggest that both head level and downstream flow rate should be treated as local states of the node. This would suggest that the dynamical model should at least be of second order. In observing the flows, however, we noticed a highly nonuniform flow profile across a pipe. In particular, we saw

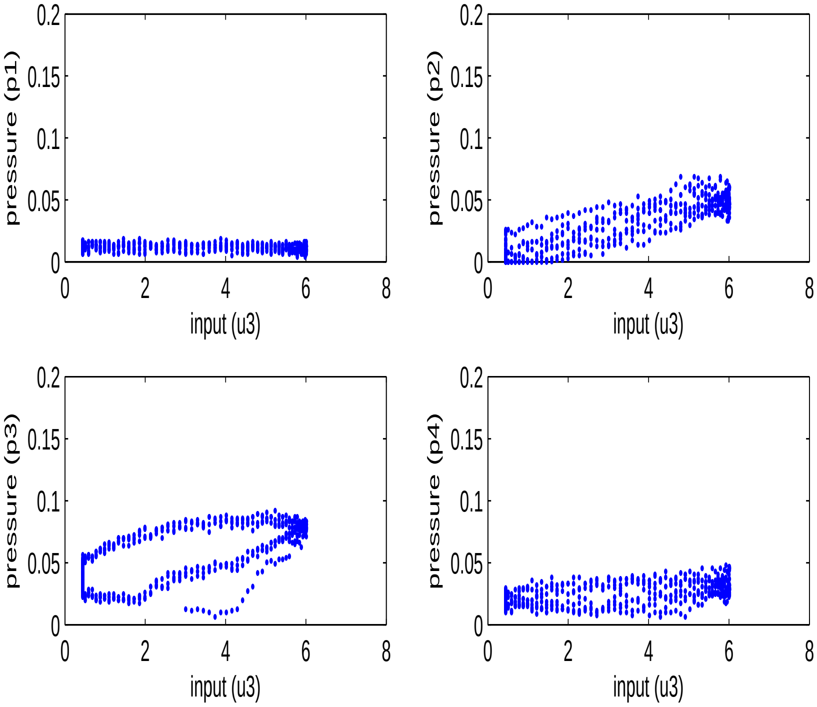


FIGURE 7. Plots of presure (p1-p4) versus control input at node 3 (u3)

that when water began backing up in the upstream pipe, there would be a lower level of rather quiet water, with most of the flow moving over the top of this layer. This observation led us to hypothesize that when the downstream flow rate was highly constrained due to frictional resistance, S_f, that we might expect an additional (third) state for the dynamics. This state would capture the energy stored in the water backed up in the upstream link of the node. These conjectures were later confirmed when we used Matlab's toolbox to identify linear dynamical models of the local node dynamics.

Dynamical models for head level dynamics were identified using Matlab's system identification tool box. The testbed was operated with a fast sinuosoidal input that was a raised sinusoid with a period of 5 minutes and a peak inflow rate of 6 gpm. The head levels (pressure measurements) for this inflow were saved and then used to identify a dynamical model for the node. A second set of "test" run data were then generated using a slower sinusoidal input that was again a raised sinusoid with a period of 15 minutes and a peak inflow rate of 6 gpm. This second run was then used to "test" the model.

To do the identification, we chose the following dynamical state space model,

$$\begin{aligned} x_i[k+1] &= A_i x_i[k] + B_i v_i[k] \\ y_i[k] &= C_i x_i[k] + D_i v_i[k] \end{aligned}$$

where

$$v_i[k] = \begin{bmatrix} \sum_{j=1}^{i} u_j[k] \\ y_{i+1}[k] \\ e_i \end{bmatrix}$$

where x_i is the state vector for node i, y_i is the water depth (pressure measurement) of node i, and u_i is the control input (diverted flow) at node i. Disturbance e_i is a white noise sequence with unit variance. The sampling period used for constructing the model was $T = 3$ seconds which is the same as the South Bend system flow simulation program.

A matlab program was written to identify a 3 order system model for the testbed data. Figure 8 shows the results of this fit. In this case, we show the identification of the second node's model for a given input at the first node. The top plot shows the identified model's prediction of the output to the training input. Whereas the bottom plot shows the predicted input to the test input. Note that both fits appear to work rather well. We did this for all of the nodes with similar performance levels, thereby suggesting that the identified models provided an accurate dynamical picture of the node's head level dynamics.

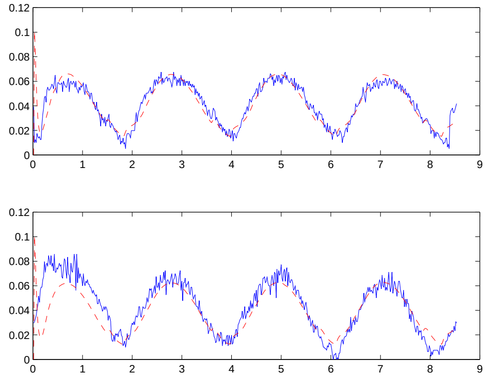


FIGURE 8. Results from identification of node 2 in the testbed

The results obtained for the testbed provided a procedure for identifying the dynamical models of the South Bend interceptor sewer. In this case, however, we used the SWMM model to generate the test and training inputs to the system. The outputs of that identification were then used to construct the required dynamical models. The identification process was done in the following way.

- **Steady state generation:** Run the South Bend flow simulation program with a specified rain input (`Q_outfall.mat`) until the system is at steady state (after $t - 6000$ steps in the program).
- **Identification data generation:** Perturb one of the nodes in the interceptor line (say node j) with two different sinusoidal signals, one with high frequency

  $$s_h(t) = M_j + M_j \sin(0.003t)$$

  and one with low frequency

  $$s_l(t) = M_j + M_j \sin(0.0008t)$$

The amplitude of the signal M_j is chosen so that none of the nodes in the interceptor line has overflow for all time. Record input $u_h(t)$ and water depth $y_h(t)$ data generated by the high frequency perturbantions $s_h(t)$. Also record input $u_l(t)$ and water depth $y_l(t)$ data generated by the low frequency perturbation $s_l(t)$.

- **System Identification:** Use $u_h(t)$ and $y_h(t)$ to identify the state models for node j's downstream nodes. Data generated by the low frequency signal is used to check the validity of the identified model. This is done by using $s_l(t)$ as the input signal to the identified system, then compare its output $y_l(t)$ with the recorded output $y_l(t)$.

The steady state generation and identification data generation is done by the flow simulation program. The flow simulation program, which is in the `csoid/soutbendsimu` directory. This program simulates the complete flow dynamics of the South Bend interceptor sewer line. The main program is `CSOsimu.m` and we can change our control law in `control_algorithm.m`.

A general interceptor model consists of $N+1$ nodes of the structure shown in figure 10. The model has N pipes where the $(N+1)$st node is directly connected to the wastewater treatment plant (WWTP). The program simulates the flow dynamics by applying a modified Euler's method on the complete dynamic wave model. The modified Euler's method approximates the solution of the initial value problem $y' = f(t,y)$ with $y(a) = y_0$ over the identified time interval $[a,b]$ at a discrete set of points using

$$\begin{aligned} y_{k+1} &= y_k h f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f(t_k, y_k)\right) \\ t_{k+1} &= t_k + h \end{aligned}$$

for $k = 0, 1, 2, \ldots.$ where h is the stepsize.

In our case the flow dynamics is simulated in the following manner.

- At time t_k $(k = 0, 1, 2, \ldots)$
  - Compute parameters $A, P_2, \ldots$ at upstream end, downstream end, and middle of each pipe at time t_k.
  - Compute half step flow $Q(t_k + h/2)$ based on $Q(t_k)$ and $H(t_k)$ (using the full momentum equation).
  - Compute the diverted flow $Q_d(t_k + h/2)$ based on $Q_{\text{outfall}}(t_k)$ and $H(t_k)$.
  - Compute $H(t_k + h/2$ based on $Q_d(t_k + h/2)$ and $\frac{1}{2}(Q(t_k) + Q(t_k + h/2))$ (using the continuity equation).
- At time $t_k + h/2$ for $k = 0, 1, 2, \ldots,$
  - Compute parameters $A, P_w, \ldots$ at time $t_k + h/2$.
  - Compute $Q(t_k + h)$ based on $Q(t_k)$ and $H(t_k + h/2)$.
  - Compute the diverted flow $Q_d(t_k + h)$ based on $Q_{\text{outfall}}(t_k + h)$ and $H(t_k + h/2)$.
  - Compute $H(t_k + h)$ based on $Q_d(t_k + h)$ and $\frac{1}{2}(Q(t_k) + Q(t_k + h))$.

The problem tries to minimize the total overflow. The control algorithm is implemented to compute the diverted flow $Q_d(t_k + h/2)$ and $Q_d(t_k + h)$.

The system identification is done by `ID.m`. In the process of identification, we noticed that perturbing one node in the line can only successfully identify a very small number of its downstream nodes. So we perturb different nodes until all the nodes have been successfully identified. Each row in the table consists of four numbers. The corresponding relationship is recorded in `id_table.mat`. The first number is the node ID we need to identify, the second one is the perturbed node ID used to do the identification, the third one is the order of the resulting linear state space system, and the fourth one is the peak value of the perturbation. An auotmated program which identifies all of the nodes at one time is `ID_all.m`, and the reuslting linear state models for all nodes are stormed in `NodeSys.mat`.

The system matrices A_i, B_i, C_i, and D_i of the linear state space models for node 1 to node 35 are in the appendix at the end of this document. Node 6 is a first order system. Nodes 26 and 35 are third-order systems, and all other nodes are second-order systems.

7. CSO Problem Statement

This section formulates the problem of controlling CSO events as an optimal control problem. The problem statement is based on our simplified model of the continuity and momentum equations. We use a municipal sewer line to illustrate the problem. The methodology, however, can be extended to other network topologies.

Many municipal sewer systems are combined sewers that mix storm and sanitary flows. The trunk lines from the combined sewers are connected to a large *interceptor sewer*. The interceptor sewer intercepts flows from the combined sewer and directs these flows to a wastewater treatment plant (WWTP) where the flows are treated and released into the environment. The combined sewer trunk line and interceptor sewer connect at a *CSO diversion structure*. The CSO diversion structure is the point where we can apply control. Many current systems use a local thresholding control. When the depth of the flow exceeds a fixed preset threshold, the flow is diverted into the interceptor sewer for subsequent treatment at the WWTP. The control strategy we propose using adjusts the amount of water diverted into the interceptor sewer based on a threshold that is a function of the system's current state.

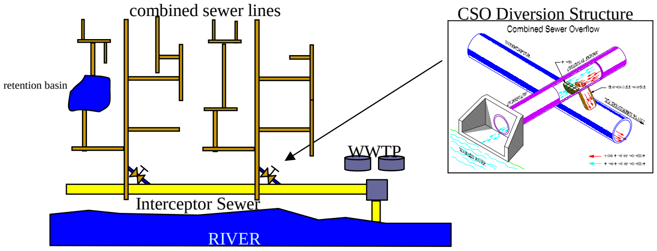


Figure 9. South Bend Interceptor Sewer and CSO Diversion Structure

The sewer system in figure 9 can be abstracted to a straight line of $N+1$ interconnected nodes, as shown in figure 10. The state of the link leaving the ith node is the flow Q_i. The state of the ith node along the line is the head level H_i for $i = 1, \dots, N$. These N nodes represent the manholes along the interceptor sewer. The $N+1$st node in the system is the WWTP, where its head level H_{N+1} is the ground level. Above each manhole node is a CSO diversion node. The flow entering this node is the external inflow w_i, the input from the old sewer lines (sanitary water, rainfall, etc). The two flows leaving each CSO diversion node are O_i the overflow dumped into the river (overflow) and u_i the flow diverted into the ith manhole node from the ith CSO diversion node. This diverted flow, u_i, represents our control variable.

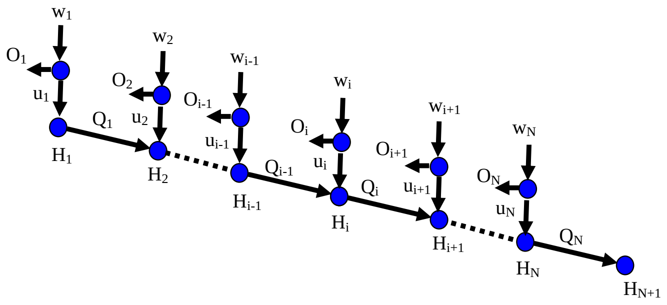


FIGURE 10. Graph of South Bend Interceptor Sewer

Our control problem seeks to minimize the total overflow from all CSO diversion nodes subject to state constraints on the nodes/links and a maximum flow limit for the entire network. Minimizing the total overflow is equal to maximizing the total diverted flow. Our problem therefore seeks to maximize

$$J(u_1, \cdots, u_N) = \sum_{i=1}^{N} \int_0^{T_s} C_i u_i(\tau) d\tau \tag{7}$$

subject to

$$\begin{aligned}
a_i(\tau)\frac{dH_i(\tau)}{dt} &= u_i(\tau) + Q_{i-1}(\tau) - Q_i(\tau) \qquad (8)\\
0 &= H_i(\tau) - H_{i+1}(\tau) - k_i(\tau)Q_i^2(\tau) \qquad (9)\\
0 &\leq u_i(\tau) \leq w_i(\tau) \qquad (10)\\
\overline{H}_i &\geq H_i(\tau) \qquad (11)\\
\overline{Q} &\geq \sum_{j=1}^{N} u_j(\tau) \qquad (12)
\end{aligned}$$

for $i = 1, \ldots, N$ and $\tau \in [0, T_s)$ where $Q_0(\tau) = 0$ and $H_{N+1}(\tau) = 0$. In equation 7, C_i is a set of weighting coefficients (costs), and T_s is the horizon (storm duration) over which to maximize the diverted flow. Equations 8-9 characterize the network's state variables. The optimization is done subject to the control being admissible (equation 10 and 12) and state constraints in equation 11. The constant $\overline{H}_i$ in equation 11 represents the head level when the manhole begins flooding. The constant $\overline{Q}$ represents a maximum flow limit for the entire network. This maximum flow limit may originate in limitations on the WWTP's capacity.

8. Supervisory Optimal Controller

Rather than working directly with the objective in equation 7, we consider the finite horizon objective in which the nth $(n = 0, \cdots, \infty)$ optimal control problem tries to maximize

$$J_n(u_1, \cdots, u_N) = \sum_{i=1}^{N} \int_{t_n}^{t_n+T_n} C_i u_i(\tau) d\tau \tag{13}$$

subject to the constraints given above in equations 8-12. In this case, T_n represents a time horizon over which the nth problem's input flows $w_i(\tau)$ are nearly constant. The time $t_n = \sum_{i=0}^{n-1} T_i$ is the inital time for the nth problem. We assume $t_0 = 0$. This section uses Pontryagin's maximum principle as a necessary characterization of "optimal" controls. This characterization is then used to generate a switching control strategy.

Assumption A1: The underlying assumption throughout this section is that the dynamics of the system in equations 8-9 are "slow". This assumption means over the interval $[t_n, t_n + T_n]$, the flow Q and storm inflows, w_i, vary slowly enough to be taken as constant.

Set of Admissible Controls: The problem's control lies in a bounded set of admissible controls. This occurs because the diverted flow, u_i, clearly cannot be larger than the storm inflow, w_i. Let $u = [u_1, \cdots, u_N] \in \Re^N$ denote the vector formed by the controls. At time $t \in \Re$ the control vector lies in an admissible set $U_t \subset \Re^N$ defined by

$$U_t = \left\{ u \in \Re^N \; : \; \begin{array}{l} \sum_{j=1}^{N} u_j \leq \overline{Q} \\ 0 \leq u_i \leq w_i(t) \quad (i = 1, \ldots, N) \end{array} \right\} \tag{14}$$

where $\overline{Q}$ is the maximum flow capacity of the network.

Without loss of generality we assume $n = 0$ in equation 13 where $t_0 = 0$. Let's assume that $T = T_0$ is chosen so that none of the state constraints in equation 11 is active. The following theorem provides a simple necessary condition that must be satisfied by the optimal control u^* over this interval.

> **Theorem 1:** Assume $\sum_{j=1}^{N} w_j(\tau) \geq \overline{Q}$ for all $\tau \in [0, T]$. Let $u^* : [0, T] \rightarrow \Re^N$ maximizes the finite-horizon objective in equation 13 subject to constraints in equations 8-12. Further assume that the constraints in equation 11 are not active under the optimal control $u^*(\tau)$ for any $\tau \in [0, T]$. Then for each $\tau \in [0, T]$, the optimal control solves the linear program
>
> $$\begin{array}{ll} \text{maximize:} & \sum_{i=1}^{N} C_i u_i(\tau) \\ \text{subject to:} & u(\tau) \in U_\tau \end{array} \tag{15}$$

Proof: The problem's control Hamiltonian $\mathcal{H} : \Re^{3N} \rightarrow \Re$ is

$$\mathcal{H}(H, p, u) = \sum_{i=1}^{N} C_i u_i(\tau) + \sum_{i=1}^{N} p_i(\tau) f_i(H, u_i) \tag{16}$$

where

$$f_i(H, u_i) \;\; = \;\; \frac{1}{a_i} (u_i + Q_{i-1} - Q_i) \tag{17}$$

and $p_i : [0,T] \to \Re$ $(i = 1, \ldots, N)$ are the problem's costates. By Pontryagin's maximum principle (PMP) we know that u^* satisfies

$$u^*(\tau) = \arg\max_{u \in U_\tau} \sum_{i=1}^{N} \left(C_i + \frac{p_i(\tau)}{a_i} \right) u_i \tag{18}$$

At each time τ, the u^* in equation 18 is the solution to a linear program. To solve this problem we need to know the costate trajectory $p(\tau)$ for $\tau \in [0,T]$. The costate satisfies the differential equation

$$\dot{p}_i = \left\{ \begin{array}{ll} E_i p_i - F_i p_{i+1} & i = 1 \\ -E_{i-1} p_{i-1} + (F_{i-1} + E_i) p_i - F_i p_{i+1} & 1 < i < N \\ -E_{i-1} p_{i-1} + F_{i-1} p_i & i = N \end{array} \right. \tag{19}$$

where $E_i = \frac{1}{2a_i k_i Q_i}$ and $F_i = \frac{1}{2a_{i+1} k_i Q_i}$. Under assumption A1 the time interval T is short so that a_i, k_i, and Q_i are constants. The costate equation is therefore a linear differential equation. Because there is no terminal penalty in the objective function in equation 13, we know that $p_i(T) = 0$ for all i. The only way this can happen is if $p_i(\tau) = 0$ for all $\tau \in [0,T]$. The linear program in equation 18 therefore has the objective $\sum_{i=1}^{N} C_i u_i$, thereby completing the theorem's proof. $\diamondsuit$

We now consider the problem in which at least one of the state constraints in equation 11 is active.

Theorem 2: Assume $\sum_{j=1}^{N} w_j(\tau) \geq \overline{Q}$ for all $\tau \in [0,T]$. Let $u^* : [0,T] \to \Re^N$ maximizes the finite-horizon objective in equation 13 subject to constraints in equations 8-12. Let Q^* denote the flows under this optimal control. Let $\Omega \subset \{1, \cdots, N\}$ and assume for all $j \in \Omega$ that the state constraints in equation 11 are active. Then for each $\tau \in [0,T]$ the optimal control solves the linear program

$$\begin{array}{ll} \text{maximize:} & \sum_{i=1}^{N} C_i u_i(\tau) \\ \text{subject to:} & 0 \leq u_i(\tau) \leq w_i \quad (i \notin \Omega) \\ & u_j(\tau) = Q_j^*(\tau) - Q_{j-1}^*(\tau) \quad (j \in \Omega) \\ & \sum_{i=1}^{N} u_i(\tau) \leq \overline{Q} \end{array} \tag{20}$$

Proof: The proof of this theorem is nearly identical to the proof in theorem 8 except that we use the version of Pontryagin's maximum principle that applies to state constraints. The costate in this case satisfy

$$\dot{p} = -\frac{\partial \mathcal{H}}{\partial x} + \sum_{j \in \Omega} \lambda_j \frac{\partial f_j(H^*, u_j)}{\partial x} \tag{21}$$

where λ_j is a Lagrange multiplier associated with the constraint $f_j(H^*, u_j) = 0$. By the maximum principle we require that u^* satisfy equation 18 subject to the constraint that $f_j(H^*, u_j) = 0$ for those nodes whose state constraint is active. Requiring $f_j(H^*, u_j) = 0$ for $j \in \Omega$, is the same as requiring $u_j = Q_j^* - Q_{j-1}^*$, which gives the second constraint in the linear program. Under assumption A1, we approximate Q^* as constant in the interval. This enable us to write $f_j(H^*, u_j) = f_j(u_j)$. In this way the second term on the right side of the costate equation is zero, and we have the exact same costate equation as in equation

19. Since we also have $p_i(T) = 0$ for all i, the rest of the proof follows the same as theorem 8, which completes the proof. ♢

Theorems 8 and 8 both suggest that $u^*(\tau)$ will lie on the boundary of the admissible set, U_τ, of controls. The following theorem shows that the optimal control must be a switching control.

> **Theorem 3:** Let $u^* : [0, T] \to \Re^N$ be the optimal control for the problem in theorem 8. If the costs C_i are all distinct, and can be ordered as $C_{i_{j+1}} < C_{i_j}$ for $j = 1, \ldots, N-1$, then for each $\tau \in [0, T]$ there exists a $r \in \{1, \ldots, N\}$ such that
>
> $$u^*_{i_j}(\tau) = \begin{cases} w_{i_j}(\tau) & 1 \le j \le r-1 \\ \overline{Q} - \sum_{k=1}^{r-1} w_{i_k}(\tau) & j = r \\ 0 & r+1 \le j \le N \end{cases} \tag{22}$$

Proof: We will use the Karush-Kuhn-Tucker (KKT) condition to prove the theorem. For simplicity, we will use u^* and w instead of $u^*(\tau)$ and $w(\tau)$ in the proof.
Define

$$\begin{aligned} L(u^*, \lambda, \mu, \nu) &= \sum_{j=1}^{N} C_{i_j} u^*_{i_j} + \lambda \left(\sum_{j=1}^{N} u^*_{i_j} - \overline{Q} \right) \\ &\quad + \sum_{j=1}^{N} \mu_{i_j}(-u^*_{i_j}) + \sum_{j=1}^{N} \nu_{i_j}(u^*_{i_j} - w_{i_j}) \end{aligned}$$

where for $j = 1, \ldots, N$

$$\begin{array}{lll} \lambda \le 0, & \sum_j u^*_{i_j} - \overline{Q} \le 0, & \lambda(\sum_j u^*_{i_j} - \overline{Q}) = 0 \\ \mu_{i_j} \le 0, & -u^*_{i_j} \le 0, & \mu_{i_j}(-u^*_{i_j}) = 0 \\ \nu_{i_j} \le 0, & u^*_{i_j} - w_{i_j} \le 0, & \nu_{i_j}(u^*_{i_j} - w_{i_j}) = 0 \end{array} \tag{23}$$

Since u^* is optimal, we have

$$\frac{\partial L}{\partial u^*_{i_j}} = C_{i_j} + \lambda - \mu_{i_j} + \nu_{i_j} = 0, \quad j = 1, \ldots, N \tag{24}$$

We first show by contradiction that the constraint

$$\sum_{j=1}^{N} u^*_{i_j} - \overline{Q} \le 0,$$

is active. Assume this constraint is not active. This means $\lambda = 0$, which implies $C_{i_j} = \mu_{i_j} - \nu_{i_j}$ for all j. By equation 23, μ_{i_j} and ν_{i_j} are either both 0 (if $0 < u^*_{i_j} < w_{i_j}$), or one is 0 and the other is nonpositive (if $u^*_{i_j} = 0$ or $u^*_{i_j} = w_{i_j}$). Since $C_{i_j} > 0$, this implies that $\mu_{i_j} = 0$, and $\nu_{i_j} < 0$. We can therefore conclude that $u^*_{i_j} = w_{i_j}$ for all j, which implies $\sum_{j=1}^{N} u^*_{i_j} = \sum_{j=1}^{N} w_{i_j} \ge \overline{Q}$. This contradicts the assumption that the constraint is not active.

We next show there exists at most one r such that $0 < u^*_{i_r} < w_{i_r}$. Assume that there exists two, say $u^*_{i_r}$ and $u^*_{i_m}$. From equation 23, we have $\mu_{i_r} = \nu_{i_r} = \mu_{i_m} = \nu_{i_m} = 0$. This means $\lambda = -C_{i_r} = -C_{i_m}$ which contradicts the assumption that the coefficients C_i are distinct.

Finally let's assume there exists one r that $0 < u^*_{i_r} < w_{i_r}$, then $\mu_{i_r} = \nu_{i_r} = 0$ so that $\lambda = -C_{i_r}$ from equation 24. Applying equation 24 to $j \neq r$, yields $C_{i_j} - C_{i_r} = \mu_{i_j} - \nu_{i_j}$. It is easily shown that if $C_{i_j} > C_{i_r}$, then $u^*_{i_j} = w_{i_j}$. Furthermore if $C_{i_j} < C_{i_r}$, then $u^*_{i_j} = 0$. This means there exists a $r \in \{1, \ldots, N\}$ such that the optimal control u^* is given by equation 22. $\diamondsuit$

Remark: The theorem says that the order of the costs C_i is the same as the order of nodes being chosen to divert flow. If we have $C_{i_{j+1}} < C_{i_j}$, then node i_1 will divert w_{i_1}, and node i_2 will divert w_{i_2}, until there comes a node i_r. Node i_r can't divert w_{i_r} because that will violate the capacity limit $\overline{Q}$, so it can only divert $\overline{Q} - \sum_{k=1}^{r-1} w_{i_k}$ to make sure the limit is satisfied. Then for $j > r$, node i_j can't divert any flow since the WWTP limit constraint is already active.

If some of the state constraints are active, as stated in theorem 8, then we will have $u^*_j(\tau) = Q^*_j(\tau) - Q^*_{j-1}(\tau)$ for $j \in \Omega$. For $j \notin \Omega$, $u^*_j(\tau)$ will follow the same as in theorem 8. This means one of them will satisfy $0 < u^*_j(\tau) < w_j(\tau)$, and the others are either 0 or $w_j(\tau)$.

If C_i are not all distinct, the optimal control u^* might not be unique. This happens when there exists a m that $C_{i_r} = C_{i_m}$, which means we can pick either node i_r or i_m to divert the amount of $\overline{Q} - \sum_{k=1}^{r-1} w_{i_k}(\tau)$, or we can share this amount of flow among them. In this case, the solution given by equation 22 is still one of the optimal controls.

Theorem 8 gives a switching control rule. For each node, the control input is either "full on" or "full off", with the exception of one node chosen to ensure the constraint $\sum_{j=1}^{N} u_j = \overline{Q}$ is active. The strategy is slightly modified when some of the state constraints are active. This modification is detailed below.

We assume that the costs C_i are all distinct, and can be ordered as $C_{i_{j+1}} < C_{i_j}$ for $j = 1, \ldots, N-1$. At a given time instant t each node i_j chooses u_{i_j}. This selection falls into one of two cases.

- In the first case, node i_j hits its state constraint $H_{i_j} = \overline{H}_{i_j}$ so that $i_j \in \Omega$. When this happens, node i_j sets its control to

$$u_{i_j} = Q_{i_j} - Q_{i_j - 1} \tag{25}$$

- In the second case, node i_j does not hit the state constraint so that $i_j \notin \Omega$. This node then selects u_{i_j} according to the following rule.

$$u_{i_j} = \begin{cases} w_{i_j} & \widetilde{Q} - W \geq w_{i_j} \\ \widetilde{Q} - W & 0 < \widetilde{Q} - W < w_{i_j} \\ 0 & \widetilde{Q} = W \end{cases} \tag{26}$$

where

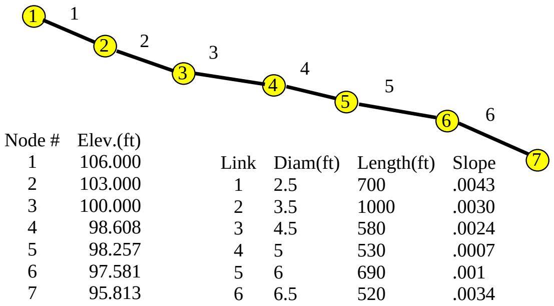
$$\widetilde{Q} = \overline{Q} - \sum_{i_j \in \Omega} (Q_{i_j} - Q_{i_j - 1}), W = \sum_{i_k \notin \Omega, k=1}^{j-1} u_{i_k} \tag{27}$$

$\widetilde{Q}$ is the total free capacity for node $i_j \notin \Omega$ and W is the total used capacity for node $i_j \notin \Omega$. Note that if Ω is an empty set, then the control rule simplifies to equation 22.

Equation 25 and 26 describe a rule that nodes use to select their control actions. We now need to characterize those time instants, t, when the control rules are applied. We can identify two distinctly different types of time instants. If node i's state constraint is active, then it will need to acquire real-time data about the flows Q_i and Q_{i-1}. This node's control would therefore need to be updated at a periodic rate which for this class of applications is on the order of $10 - 30$ seconds. The other type of decision time occurs at nodes that have not hit their state constraints. The control decision made by these nodes is essentially a switching decision (open or close the valve) and this only needs to change when the "discrete-state" (i.e. the set of active state constraints Ω) or the storm inflow w changes. In practice, the storm inflows and state constraint set Ω change slowly over time. This set of decision times tends to be a low rate aperiodic stream of times.

9. Simulation Results for "Optimal" Control

This section presents simulation results of our controller on a high-fidelity model of the 7-node interceptor sewer shown in figure 11 and the 36 node interceptor sewer of the city of South Bend, Indiana.



Node #	Elev.(ft)
1	106.000
2	103.000
3	100.000
4	98.608
5	98.257
6	97.581
7	95.813

Link	Diam(ft)	Length(ft)	Slope
1	2.5	700	.0043
2	3.5	1000	.0030
3	4.5	580	.0024
4	5	530	.0007
5	6	690	.001
6	6.5	520	.0034

Figure 11. 7-node Interceptor Sewer used in Simulation

In this simulation, the parameters are listed in the following table. For the passive thresholding strategy, the fixed threshold is 40 (we only allow a maximum $40(cfs)$ flow into the interceptor line at each diversion structure). The external inflow is a rain event that last for 12 hours. The $40(cfs)$ threshold was chosen so that if all lines into the interceptor sewer exceeded this rate, then the interceptor sewer would flood. This is a commonly used criteria by which many real-life water districts set their CSO thresholds.

N	$T_s(min)$	$\overline{Q}(cfs)$	C
6	720	280	$[1.5, 1.4, 1.3, 1.2, 1.1, 1]^T$
$\overline{H}(ft) = [108.50, 105.50, 103.50, 103.11, 103.26, 103.58]^T$			

TABLE 1. Simulation Parameters

Figure 12 plots the head levels of node 2 and 3 as a function of time (The dotted lines are the corresponding $\overline{H}_2$ and $\overline{H}_3$). It shows that the head constraints $H_2(t) \leq \overline{H}_2$ and $H_3(t) \leq \overline{H}_3$ are satisfied for all t, while each of them becomes active over some horizon. This means no flooding occurs during the rain event (Head levels at other nodes are always below the maximum level, and are not plotted here).

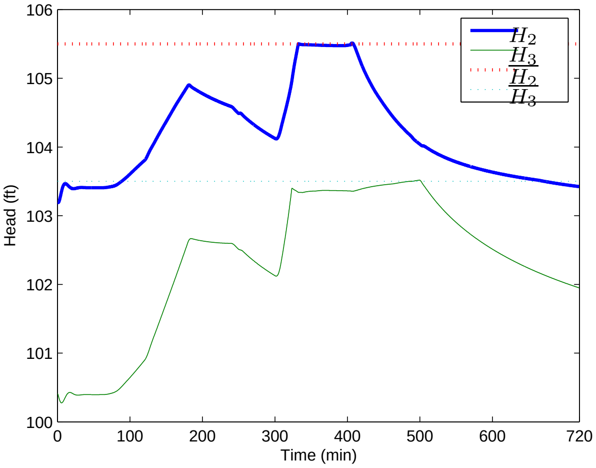


FIGURE 12. Head level when using our control

Figure 13 has five curves on it. They plot the time history of different flows. The one on top is the total rain flow. The straight line below it represents the WWTP capacity limit. The third curve, which lies right below the straight line, is the flow (Q_6) into the WWTP when our control is used. As we can see, the flow is always less or equal to the WWTP capacity limit. The fourth and fifth curve represent the total overflow when the passive thresholding strategy and our control is applied, respectively. The overflow under our control is significantly less than the other case. When the passive thresholding strategy is used, the total overflow is $2.6061 \times 10^6 (ft^3)$. The overflow drops to $5.4549 \times 10^5 (ft^3)$ when our control is applied. This is a decrease of 79.1%.

The algorithm was also simulated on the real South Bend interceptor sewer line which consists of 36 CSO diversion structures. The following three different storm scenarios are considered. Each storm drops rain nonuniformly over the city and moving from west to east over city at 20mph.

- S1. 0.485 inch of rain in 11 hours
- S2. 0.799 inch of rain in 13 hours
- S3. 2.046 inch of rain in 19 hours

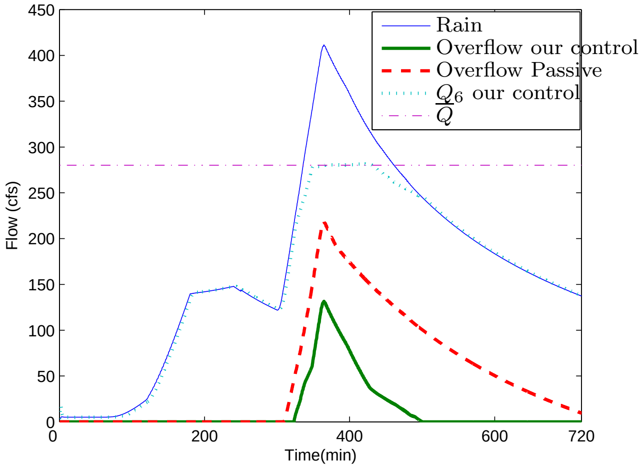


Figure 13. Overflow comparison using our control vs passive thresholding

The simulation results are given in table 2. It shows the existing system overflow (using fixed thresholding strategy), controlled system overflow (when implementing our control scheme), and overflow decrease in percentage obtained by our control scheme over South Bend's existing CSO system. You can see from the table that our proposed approach reduces the total storm overflow by $24\% - 40\%$, which is significant.

Storm	existing overflow $(ft^3 \times 10^6)$	controlled overflow $(ft^3 \times 10^6)$	overflow decrease
S1	0.46	0.28	40%
S2	2.51	1.90	24%
S3	6.04	3.79	37%

Table 2. Overflow comparison

10. Local Pressure Measurement Feedback Control

If a node's flooding constraint is inactive, then the Wan controller only requires each node only needs to know whether or not there is sufficient capacity left in the system for it to open up its diversion valve. This type of regulation is a supervisory decision that requires little in the way of real-time feedback. When a node's "flooding" constraint becomes active, however, the node must locally control its diverted flow to guarantee that the head level constraint is not exceeded. As noted in the preceding sections, this type of control requires real-time measurement of the flows in the upstream and downstream links.

The problem we have with such a controller, however, is that sensors providing flow measurements are inaccurate and expensive. The inaccuracy of these sensors arises from the fact that we are trying to determine an "average" flow rate using point sensor in a turbulent flow. A more reliable measurement of the node's local state is given by a pressure sensor that can be used to directly infer the head level in the manhole. As it turns out, such pressure sensors are more accurate and less expensive than the flow rate sensors. We must therefore modify our flow control strategy so we can still guarantee no flooding using only pressure (or rather head level) measurements. The output feedback controller would only be activated when the local node's flooding constraint becomes active. This section discusses the technique that was used to develop such a controller. In particular, we will show that this controller is a *decentralized controller* that only requires "local" measurements of the head level to maintain overall system stability.

The development of a feedback controller requires a previously identified dynamical model for the node's input/output behavior. The identification of such models was discussed in section 6. We now review that model as it pertains to the development of the output feedback controller.

The ith node's head level (pressure level) is denoted as y_i. We assume that y_i is generated by a discrete-time state based model of the form

$$\begin{aligned} x_i[k+1] &= A_i x_i[k] + B_{1i} \sum_{j=1}^{i} u_j[k] + B_{2i} y_{i+1}[k] + B_{3i} e_i[k] \\ y_i[k] &= C_i x_i[k] + D_{1i} \sum_{j=1}^{i} u_j[k] + D_{2i} y_{i+1}[k] + D_{3i} e_i[k] \end{aligned}$$

where $x_i[k]$ is the node's state, $u_i[k]$ is the diverted flow (our control) for the ith node and e_i is a white noise input. These models were previously identified from the South Bend Interceptor sewer system using the modified SWMM model (see appendix). In that case, we can express the dynamics for the ith node using as

$$G_i \overset{s}{=} \left[\begin{array}{c|ccc} A_i & B_{1i} & B_{2i} & B_{3i} \\ \hline C_i & D_{1i} & D_{2i} & D_{3i} \end{array} \right]$$

From the appendix, we see that $D_{1i} = D_{2i} = 0$ for all systems. The input $\sum_{j=1}^{i} u_j[k]$ is the sum of all diverted flows upstream of node i. We can break this sum apart as

$$\sum_{j=1}^{i} u_j[k] = u_i[k] + \sum_{j=1}^{i-1} u_j[k]$$

where the first term represents that part that is directly controlled by node i and the second term can be treated as an external input over which node i has no direct control. In a similar way the third term is also an exogenous term representing sensor noise and unmodeled loads on the system. The other input, entering through the B_{2i} matrix, models coupling between adjacent nodes in the interceptor line.

We assume the output feedback controller for the ith node accepts y_i (pressure/head measurement) as input and output the control action (diverted flow) u_i. The controller's output u_i is generated by a discrete-time state-based system of the form,

$$\begin{aligned} \hat{x}_i[k] &= \hat{A}_i \hat{x}_i[k-1] + \hat{B}_i \left(y_i[k] - \overline{y}_i \right) \\ u_i[k] &= \hat{C}_i \hat{x}_i[k] + \hat{D}_i \left(y_i[k] - \overline{y}_i \right) \end{aligned}$$

where $\hat{x}_i$ is the ith controller's internal state and $\overline{y}_i$ is the desired reference point (flooding level) for node i. As usual, we can represent the controller's state-based realization as

$$K_i \overset{s}{=} \left[\begin{array}{c|c} \hat{A}_i & \hat{B}_i \\ \hline \hat{C}_i & \hat{D}_i \end{array}\right]$$

Note that this controller is *decentralized*. It only makes use of the local head/pressure measurements. The controller K_i must be selected to reject the disturbance vector $\begin{bmatrix} \sum_{j=1}^{i-1} u_j[k] \\ y_{i+1}[k] \\ e_i[k] \end{bmatrix}$ at the output variable $\tilde{y}_i[k] = y_i[k] - y_{i0}$. The is a somewhat standard disturbance rejection problem that can be solved using the $\mathcal{H}_\infty$ controller synthesis framework.

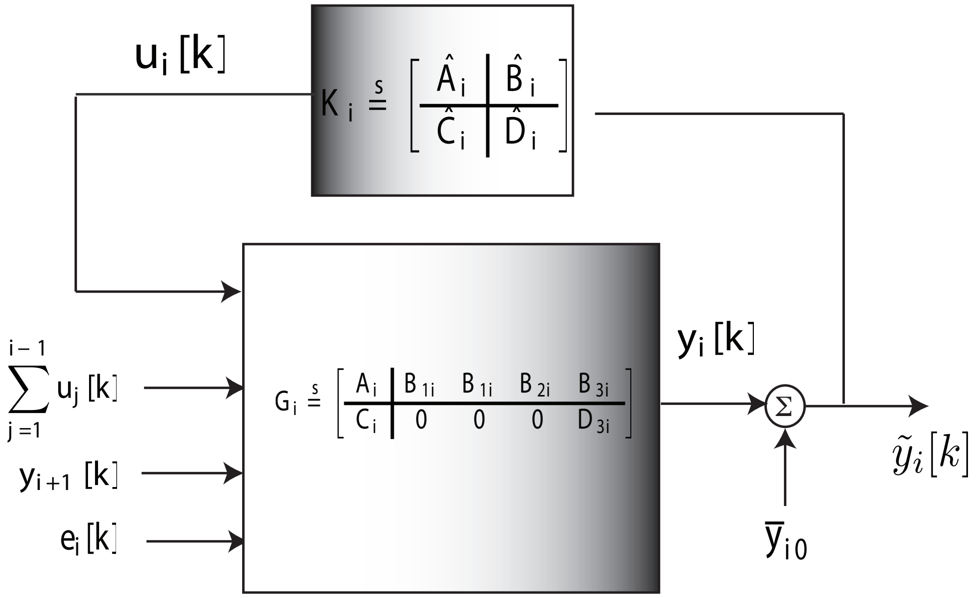

$u_i[k]$
$K_i \overset{s}{=} \left[\begin{array}{c|c} \hat{A}_i & \hat{B}_i \\ \hline \hat{C}_i & \hat{D}_i \end{array}\right]$
$G_i \overset{s}{=} \left[\begin{array}{c|cccc} A_i & B_{1i} & B_{1i} & B_{2i} & B_{3i} \\ \hline C_i & 0 & 0 & 0 & D_{3i} \end{array}\right]$
$\sum_{j=1}^{i-1} u_j[k]$
$y_{i+1}[k]$
$e_i[k]$
$y_i[k]$
Σ
$\overline{y}_{i0}$
$\tilde{y}_i[k]$

Figure 14. Block Diagram of Local Output Feedback Controller

There are many ways of formulating the $\mathcal{H}_\infty$ controller. The most commonly used approach poses this as a mixed sensitivity problem. The solution is an observer-based controller whose gains are determined by solving a pair of associated algebraic Riccati equations. The problem with this approach is that it requires a set of well-posed weighting systems to properly balance the objectives of performance maximization subject to plant uncertainty. Moreover, a straightforward application of the $\mathcal{H}_\infty$ synthesis method would usually not include any integral action, so the system may not necessarily have zero steady-state errors to step disturbances. In our case, we expect these external disturbances $\sum_{j=1}^{i-1} u_j[k]$ and $y_{i+1}[k]$ to be step inputs. So it becomes imperative that we adopt a controller that has integral action.

Rather than using the state-based $\mathcal{H}_\infty$ controller synthesis method, we chose to use a traditional loopshaping design on an equivalent continuous-time model of the discrete-time plant. Loopshaping provides a particularly easy and intuitive way of designing feedback controllers for signal-input single-output (SISO) plants that can readily include integral action. These graphical methods provide an extremely convenient way of visualizing the precise tradeoffs we need to make between disturbance rejection and robustness to modeling error. As a result of these considerations, we chose to use classical loopshaping methods to determine the local feedback controllers for this system.

We use a single node (node 8) in the South Bend interceptor sewer sytem, to illustrate the design approach used in synthesizing the local controller. Classical loopshaping is a frequency-based design method for continuous-time plants. To apply these techniques to the discrete-time system models, we first convert the discrete-time plant into a continuous-time plant using one of the standard transformation techniques. In particular, we assumed the discrete-time system was obtained by using a zero-order hold (ZOH) on a continuous-time system that was sampled once every 3 seconds. This conversion can be automated using the Matlab command `d2c`. In our case, the transfer function for the discrete-time system (node 8) was

$$\begin{aligned} G_8(z) &= \left[\begin{array}{ccc} \frac{0.00078336(z-0.9959)}{z^2-1.994z+0.9943} & \frac{0.00071813(z-0.9952)}{(z^2-1.994z+0.9943} & \frac{7e-5(z-0.3077)(z+0.5677)}{z^2-1.994z+0.9943} \end{array} \right] \\ &= \left[\begin{array}{ccc} G_{81}(z) & G_{82}(z) & G_{83}(z) \end{array} \right] \end{aligned}$$

where G_{81} maps $\sum_{j=1}^{i} u_i$ onto y_i, G_{82} maps y_{i+1} onto y_i and G_{83} maps e_i onto y_i. We then convert this to a continuous-time transfer function matrix of the form and overbound this by a transfer function

$$G_8(s) = 15\frac{0.0002613s + 3.578e-7}{s^2 + .001905s + 3.343e-5}$$

whose gain-magnitude plot is shown in figure 15. We then apply the loopshaping synthesis procedure to this loop shape.

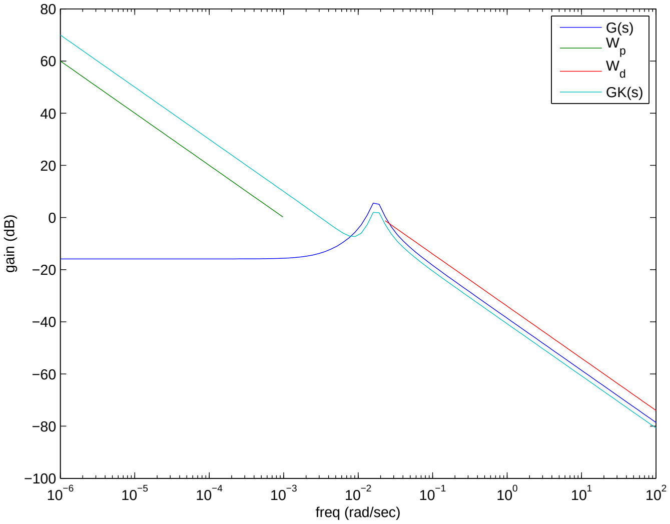


FIGURE 15. Gain-magnitude of $G_8(s)$

Loopshaping requires us to select a suitable performance weighting system $W_p(s)$ and robust stability weighting system $W_d(s)$. We then introduce a controller $K(s)$ such that the gain-magnitude of GK lies about $|W_p(s)|$ for frequencies where $|GK(s)| \gg 1$. We also require that the reshaped loop lie below $|W_d^{-1}(s)|$ for frequencies where $|GK(s)| \ll 1$ and finally we require a sustained 20 dB/decade roll off in the region of the gain-crossover frequency. In our case, we usually chose $W_p(s) = \frac{C}{s}$ where C was a constant that was chosen as large as possible. $W_d^{-1}(s)$ was also chosen to be of the same form as an integrator. For node 8 the choices were $W_p(s) = \frac{1.e{-}03}{s}$ and $W_d(s) = 50s$. These weighting systems are shown in figure 15. If we use a controller of the form

$$K(s) = \frac{11.74s^2 + 0.1204s + 0.00121}{s^2 + 0.004108s}$$

We can reshape the loop function so that GK has the gain magnitude shown in figure 15. This controller is essentially a PID controller. We then use `d2c` to convert $K(s)$ into a discrete-time system which is what we actually use to control the node. In this case, the resulting controller has the form

$$K(z) = \frac{36.21z^2 - 72.07z + 35.86}{z^2 - 1.996z + .9959}$$

The loopshaping method outlined above is the way in which all controllers for the South Bend interceptor line were designed. The resulting discrete-time controllers are itemized (in state-based form) in the appendix.

Figure 16 shows the response of the controlled system to inputs that were drawn from a simulation run of the South Bend interceptor sewer. This figure shows the head level within the sewer system increasing until it activates the head constraint around 3500 minutes into the simulation, after which the pressure-based feedback controller is able to hold the head level at the desired saturation point of 683.55. After 6000 minutes, the inflow into this node begins decreasing so that the head level also begins decreasing.

11. Distributed CSOnet Control Algorithm

The prior sections discussed the development of a supervisory control scheme that showed the optimal "control" strategy is to open up the most "important" nodes valves all of the way until a flooding constraint is activated, after which we introduce a control to prevent the flooding constraint from being violated. In the last section, we showed that this flooding constraint could be met using a decentralized pressure feedback controller. The ability of the local controller to prevent flooding, however, requires that there sufficient control authority at the local node. In other words, the control u_i must remain admissible (i.e. $0 \leq u_i \leq w_i$). If the node is at its flooding level and u_i has fallen to zero, then there is no remaining control "authority" (short of pumping water out of the manhole) to ensure that the flooding constraint is not violated. Such a situation is shown in figure 17.

The plots in figure 17 shows the time history of the head levels generated in the South Bend interceptor line under a storm scenario provided by EmNET LLC. In this scenario, inflows into node 10 activate the flooding constraint. However, there is no inflow into node 11. The inflow from node 10 (even at its flooded level) is large enough to cause node 11 to hit its flooding constraint. However, because there is no water entering node 11 from

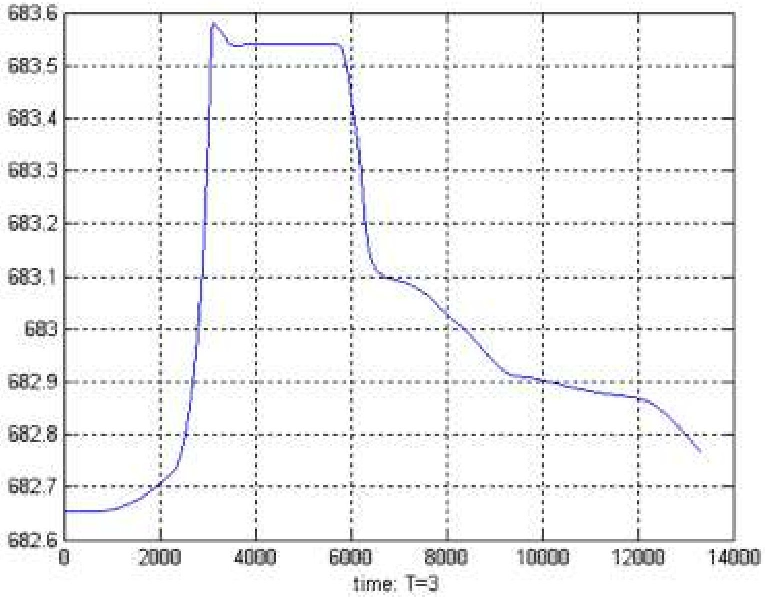


FIGURE 16. Response of Pressure-based Controller to standard input storm profile (Node 8)

the trunk line, we are unable to control the head level using node 11's throttle valve. As a result node 11 through 13 end up violating their flooding constraints.

This section presents a simple modification to the local control algorithm that allows us to meet the head level constraints while still trying to minimize the total overflow. In particular, recall that the continuity equation at node i is

$$\frac{dH_i}{dt} = u_i + Q_{i-1} - Q_i$$

Clearly the amount of diverted flow into the ith node cannot be less than zero, so if the controller commands a negative flow rate, we see that u_i actually gets set to zero and there is nothing that we can do to try and force $\frac{dH_i}{dt}$ to be zero.

One solution is to simply "freeze" the upstream value Q_{i-1} when

- node i is about to flood
- and node i's control authority, u_i, is nearly zero.

By keeping Q_{i-1} constant, we expect Q_i to also reach a steady state constant value that will be equal to Q_{i-1}. This action prevents the ith node's head level H_i, from getting any higher. By the constitutive relation,

$$H_{i-1} - H_i = kQ_{i-1}^2$$

If H_i is constant, then we can keep Q_{i-1} constant by simply controlling the upstream node's head level to freeze its value. In other words, we reset the head level constraint, $\overline{H}_{i-1}$, for the upstream node to equal its current value. This then will essentially stop the increase in node i's head level.

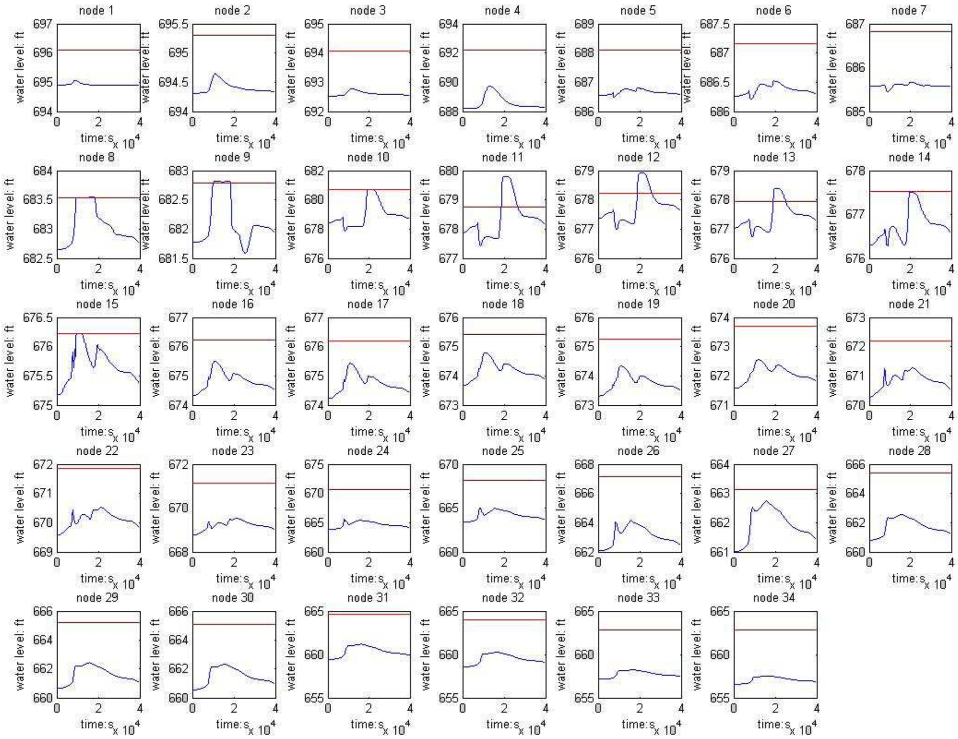


FIGURE 17. Example where the "optimal" control strategy fails to enforce the flooding constraint

So a simple switching control scheme can be adopted when the ith node is flooded and about to lose its control authority. In particular, let T_i denote the time instant when node i is about to flood and $u_i(T_i) \approx 0$. We simply have node i send a message to node $i-1$ that asks the upstream node to reset its flood constraint to $\overline{H}_{i-1} = H_{i-1}(T_i)$. In other words, the constraint is set to the $i-1$st node's head level when the flooding event occurred at node i. Node $i-1$ is therefore activating its flooding constraint which causes it to control its head level so that H_{i-1} remains less than or equal to $\overline{H}_{i-1}$. Of course, if the $i-1$st node loses its command authority then it will send a message to node $i-2$ requiring that node to reset its flooding constraint.

This modification to the control algorithm was implemented on the South Bend Interceptor sewer simulation for a number of storm that were generated by EmNET LLC. Figure 18 shows the head level time histories for this modified controller. What we can see here, of course, is that none of the flooding constraints are violated.

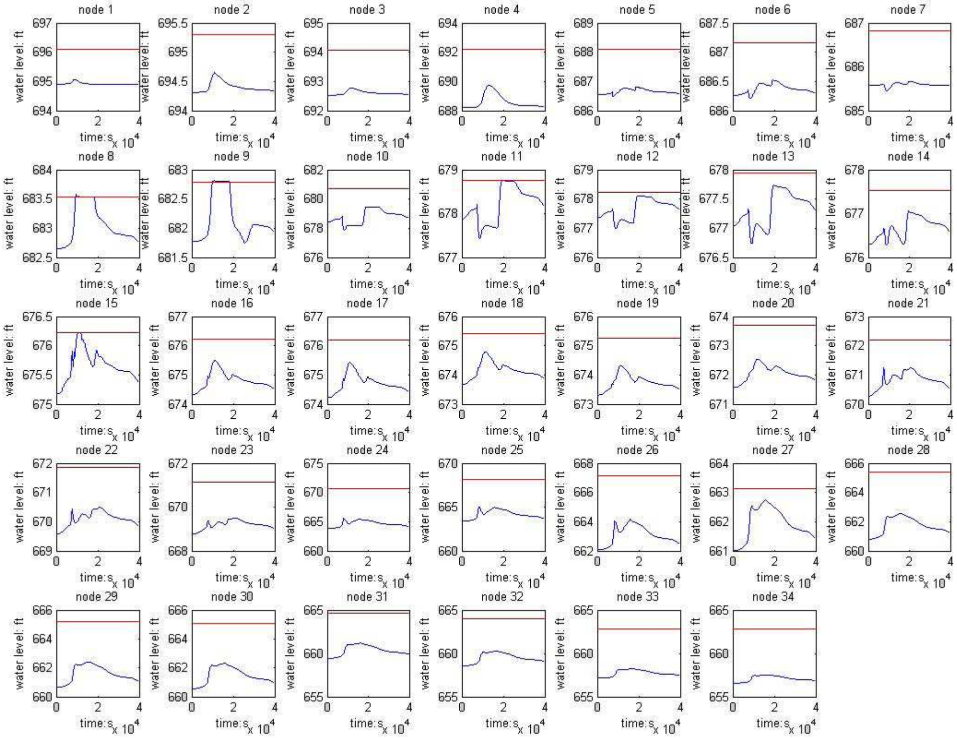


FIGURE 18. Head level time histories for modified CSOnet controllers

The proposed approach for addressing the flooding issue will clearly be suboptimal with respect to the total overflow. This is because we are essentially freezing the diverted flows in the upstream nodes before those nodes have reached their full capacity. This observation is born out in some of the simulation studies whose results are tabulated in table 3. This table shows the total overflows for 5 different storm scenarios that were provided by EmNET LLC. These overflows were computed using three different control algorithms; a passive threshold strategy, algorithm 1 and 5. Algorithm 1 was the original supervisory control algorithm in Wan et al.'s paper [7]. Algorithm 5 was the modified local control scheme described above. Note that Algorithm 1 generally resulted in slightly lower total overflow volume than the modified local algorithm described here. The increase, however, is relatively small for most of the storms.

The last column in table 3 shows the percent change in the total overflow between the passive thresholding strategy and algorithm 5. The passive thresholding strategy is what is currently implemented in the South Bend Interceptor lines. It includes overflows from multiple trunk lines that feed into the same diversion structure. In algorithm 5 (and 1), we assume that the passive weirs in these trunklines have been removed so that all of the trunkline flows can be directed into the interceptor sewer. The actual flow that is diverted into the interceptor is then determined using the decentralized pressure-based controllers using the flood prevention protocol discussed above. The last column shows the percentage change in the total overflow between the passive strategy and the decentralized controlled approach. For the smallest storm (C), we see a 60 percent reduction in total overflow. For the medium sized storm (D and E) we see a 20-26 percent reduction in total overflow. For

total overflow volume:ft^3	passive	algorithm 1	algorithm 5	Δ
storm C	405980	123750	152490	60%
storm D	1206900	770430	883560	26%
storm E	2682800	2050200	2141200	20%
storm G	9280600	8068800	8413400	9%

TABLE 3. total overflow volume

the large storm (E), we see a smaller 10-13 percent reduction. No flooding was observed in any of the algorithm 5 simulation runs. Whereas significant flooding was observed for some of the storms under the passive thresholding strategy. These results seem to indicate that the proposed CSOnet controller will meet its objectives in significantly reducing total CSO overflows while preventing localized flooding in the system and preventing diverted flows in exceedance of the wastewater treatment plant's capacity.

12. SUMMARY

This document is an interim technical report describing the current status of the CSOnet control algorithms being developed by the University of Notre Dame for EmNET LLC. The algorithms use a decentralized decision rule for opening and closing throttle valves in the CSO diversiion structures which satisfy certain necessary conditions for optimality. These decision rules open up the "most important" nodes' valves until a flooding constraint is activated. Once the flooding constraint is activated a decentralized local controller is used to prevent the head level from getting any higher. This local controller consists of two parts.

(1) There is a output feedback controller that uses local measurements of pressure (or head) to actuate the diverted flow into the interceptor line. The control is decentralized in that it only needs local measurements of the node pressure to control that node's diverted flow.
(2) There is a simple switching logic to prevent flooding when the local controller loses "control authority". In this case, if a node is about to flood and has very little remaining control authority (i.e. current diverted flow is nearly zero), then the node signals to its upstream node to reset its flood level constraint. The upstream node's flood level constraint is set equal to the upstream node's current head level. This effectively forces the flooding constraint at the upstream node to become active.

This controller was tested on a detailed model of the South Bend Interceptor sewer system for a suite of storm profiles that were provided by EmNET LLC. The results showed that the system was able to substantially reduce CSO overflows (from 10-60 percent) over the existing passive thresholding scheme used in the South Bend system. This was done while preventing localized flooding and preventing exceedance of the WWTP's capacity.

As part of this work, we built and tested a hardware testbed of an interceptor sewer. We also developed local dynamical models and controllers for each node in the South Bend Interceptor Sewer system. We anticipate that these models and controllers would be used to develop the actual CSOnet controllers that should be installed in the South Bend system by summer 2009.

The models and controllers represent the main deliverables to EmNET LLC. For the next year, there may be additional work depending in large part on what EmNET LLC requires of us. We anticipate that we will need to provide technical support when EmNET ports the proposed control algorithms to the CSOnet system. We recommend that this port first be done on the hardware testbed that was delivered earlier to EmNET. Once this is done, it should be possible to begin porting the controllers to the entire system in a systematic and phased manner.

13. Appendix A: South Bend Interceptor Sewer Models

$$G_1 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9987 & 0.00398 & 0.002556 & 0.00625 & 0.003448 \\ -0.004752 & 1 & 0.003044 & 0.004422 & 0.0701 \\ \hline 0.5152 & -0.02063 & 0 & 0 & 0.0001889 \end{array}\right]$$

$$G_2 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9965 & 0.00172 & 0.01576 & 0.01484 & 0.003232 \\ -0.005616 & 1 & 0.006145 & 0.4088 & 0.4944 \\ \hline 0.1586 & -0.0001295 & 0 & 0 & 0.0002531 \end{array}\right]$$

$$G_3 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9977 & -0.000392 & -0.001615 & 0.0106 & 0.00033 \\ -6.534e-5 & 0.9955 & -0.1052 & 0.09689 & -0.2277 \\ \hline 0.2228 & -0.0001222 & 0 & 0 & 5.74e-5 \end{array}\right]$$

$$G_4 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9985 & -0.0006864 & 0.01878 & -0.00391 & 0.0001388 \\ 0.01116 & 0.9976 & -0.01309 & -0.3118 & -0.1045 \\ \hline 0.08321 & -2.841e-5 & 0 & 0 & 1.021e-5 \end{array}\right]$$

$$G_5 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9928 & -0.01116 & 0.02993 & -0.01738 & 0.0001289 \\ -0.003601 & 0.992 & 0.01995 & -0.01552 & -0.005143 \\ \hline 0.2298 & -0.001143 & 0 & 0 & 2.693e-5 \end{array}\right]$$

$$G_6 \overset{s}{=} \left[\begin{array}{c|ccc} 0.9998 & 0.002495 & -0.004153 & 0.0001816 \\ \hline 0.1932 & 0 & 0 & 7.017e-5 \end{array}\right]$$

$$G_7 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9985 & -0.006966 & -0.01953 & 0.07254 & 0.0004989 \\ 0.01106 & 0.9722 & -0.13 & 0.283 & 0.02967 \\ \hline 0.1102 & -0.0003584 & 0 & 0 & 4.987e-5 \end{array}\right]$$

$$G_8 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9899 & 0.002281 & 0.01245 & 0.1142 & 0.002446 \\ -0.04775 & 1.004 & 0.04581 & 0.4531 & 0.3845 \\ \hline 0.06315 & -6.403e-5 & 0 & 0 & 7.447e-5 \end{array}\right]$$

$$G_9 \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9968 & 0.002462 & -0.01294 & 0.1173 & 0.00268 \\ -0.002508 & 0.9882 & 0.09462 & -0.4266 & 0.02459 \\ \hline 0.05697 & -0.0001467 & 0 & 0 & 9.194e-6 \end{array}\right]$$

$$G_{10} \overset{s}{=} \left[\begin{array}{cc|ccc} 1 & 0.0164 & 0.2389 & 0.2382 & -0.002244 \\ -0.0174 & 0.995 & 0.07429 & 1.008 & 0.2233 \\ \hline 0.01212 & 0.0005833 & 0 & 0 & 5.91e-5 \end{array}\right]$$

$$G_{11} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9869 & -0.01276 & -0.537 & -6.938 & 0.1923 \\ -0.02065 & 0.9769 & -0.9427 & -11.91 & 0.2138 \\ \hline 0.006692 & -0.005817 & 0 & 0 & 2.603e-5 \end{array}\right]$$

$$G_{12} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9983 & -0.01428 & 0.6437 & 20.37 & -0.03156 \\ -0.00375 & 0.9508 & 2.132 & 69.65 & -0.3122 \\ \hline 0.003387 & -0.0004164 & 0 & 0 & 1.389e-5 \end{array}\right]$$

$$G_{13} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9887 & 0.008521 & -0.01935 & 1.307 & 0.1512 \\ -0.01781 & 1.002 & -0.131 & -1.736 & 0.179 \\ 0.00754 & -0.006176 & 0 & 0 & 2.594e-5 \end{array}\right]$$

$$G_{14} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9588 & -0.01129 & -0.06721 & -4.147 & 0.03958 \\ -0.03503 & 0.9867 & -0.0724-3.791 & 0.03019 & \\ \hline 0.02672 & -0.03419 & 0 & 0 & 1.959e-5 \end{array}\right]$$

$$G_{15} \overset{s}{=} \left[\begin{array}{cc|ccc} 1.006 & 0.01201 & 0.03346 & 0.3125 & 0.007696 \\ -0.009632 & 0.9814 & -0.05309 & -0.4235 & 0.006675 \\ \hline 0.008514 & -0.00864 & 0 & 0 & 6.159e-6 \end{array}\right]$$

$$G_{16} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9681 & -0.00467 & 0.5655 & 66.53 & 0.08832 \\ 0.009247 & 0.9748 & 0.8033 & 80.43 & -0.1643 \\ \hline 0.004108 & -0.002324 & 0 & 0 & 2.202e-5 \end{array}\right]$$

$$G_{17} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9943 & -0.006687 & -0.01928 & -1.024 & 0.0703 \\ -0.005612 & 0.9812 & -0.05191 & -2.348 & 0.0756 \\ 0.01379 & -0.01242 & 0 & 0 & 2.566e-5 \end{array}\right]$$

$$G_{18} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9897 & -0.004651 & -0.005744 & -0.1788 & -0.000177 \\ -0.007651 & 0.979 & -0.0248 & -0.5349 & 0.03274 \\ \hline -0.1568 & 0.001427 & 0 & 0 & 5.782e-5 \end{array}\right]$$

$$G_{19} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9862 & -0.02492 & -0.09291 & -2.925 & 0.178 \\ -0.01552 & 0.963 & -0.1282 & -4.039 & 0.209 \\ \hline 0.1042 & -0.08704 & 0 & 0 & 0.0002822 \end{array}\right]$$

$$G_{20} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9708 & -0.003242 & -0.05892 & 8.087 & -0.1797 \\ -0.02486 & .994 & -0.05857 & 7.661 & -0.1853 \\ \hline 0.03981 & -0.03888 & 0 & 0 & 3.902e-5 \end{array}\right]$$

$$G_{21} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9927 & 0.0005891 & -0.1752 & 6.246-0.3319 & \\ -0.000569 & 0.994 & -0.2218 & 7.574-0.4225 & \\ \hline 0.03424 & -0.02695 & 0 & 0 & 3.181e-5 \end{array}\right]$$

$$G_{22} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.993 & 0.003367 & 0.005487 & 0.5277 & 0.003893 \\ 0.005295 & 0.9812 & -0.0214 & -2.326 & 0.1442 \\ \hline 0.04618 & -0.0007038 & 0 & 0 & 7.004e-5 \end{array}\right]$$

$$G_{23} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.989 & 0.003898 & 0.002253 & 0.4648 & 0.001331 \\ 0.003413 & 0.9813 & -0.002459 & -1.784 & 0.06498 \\ \hline 0.07759 & -0.0004276 & 0 & 0 & 6.696e-5 \end{array}\right]$$

$$G_{24} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9876 & -0.01093 & 1.427 & 144.6 & -0.0592 \\ 0.1197 & 0.9103 & 9.34 & 876.2 & -0.8143 \\ \hline 0.004259 & -0.0004113 & 0 & 0 & 8.51e-5 \end{array}\right]$$

$$G_{25} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9482 & -0.01816 & 1.08 & 38.26 & -0.3875 \\ -0.03007 & 0.9873 & 0.6538 & 22.9-0.2726 & \\ \hline 0.01047 & -0.01525 & 0 & 0 & 7.877e-5 \end{array}\right]$$

$$G_{26} \overset{s}{=} \left[\begin{array}{ccc|ccc} 0.997 & 0.009345 & -0.001772 & 0.2042 & 10.12 & -0.07266 \\ -0.005296 & 0.9823 & 0.001552 & -0.2103 & -9.658 & -0.001521 \\ 0.0009815 & 0.006619 & 0.9784 & 0.9952 & 51.92 & -0.3128 \\ \hline -0.04602 & 0.001476 & 0.01045 & 0 & 0 & 3.746e-5 \end{array}\right]$$

$$G_{27} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9749 & -0.003017 & 0.001107 & 0.821 & 0.001808 \\ -0.002691 & 0.9728 & 0.005608 & 4.712 & -0.07426 \\ \hline 0.09016 & -0.0006347 & 0 & 0 & 0.0002878 \end{array}\right]$$

$$G_{28} \overset{s}{=} \left[\begin{array}{cc|ccc} 1 & -0.03246 & 0.0002678 & 0.2356 & 0.0005449 \\ 0.012 & 0.988 & -0.0002775 & -0.06747 & -0.0424 \\ \hline 0.07641 & -0.004809 & 0 & 0 & 0.0001417 \end{array}\right]$$

$$G_{29} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9664 & 0.00292 & -0.002261 & -1.478 & 0.0377 \\ 0.05324 & 0.9718 & 0.006352 & 3.591 & 0.002098 \\ \hline 0.002655 & 0.02007 & 0 & 0 & 0.0001225 \end{array}\right]$$

$$G_{30} \overset{s}{=} \left[\begin{array}{cc|ccc} 1.003 - 0.0151 & 0.281165.66 & 0.003573 & & \\ 0.01583 & 0.9842 & 0.168 & 46.23 & -0.08947 \\ \hline 0.0004718 & -7.638e-7 & 0 & 0 & 1.328e-6 \end{array}\right]$$

$$G_{31} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9857 & 0.003429 & 0.01489 & 0.6054 & -0.2543 \\ -0.002098 & 0.996 & 0.006733 & 0.1751 & -0.2246 \\ \hline 0.03415 & -0.03927 & 0 & 0 & 0.0001081 \end{array}\right]$$

$$G_{32} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9877 & 0.01846 & -0.25 & 35.89 & 0.006096 \\ -0.01089 & 1.003 & 0.1196 & 43.61 & 0.1163 \\ \hline 0.0002951 & -2.979e-6 & 0 & 0 & 8.364e-7 \end{array}\right]$$

$$G_{33} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.855 & -0.1183 & 0.07799 & 6.262 & 0.009544 \\ 0.1866 & 1.132 & -0.07244 & -13.8 & -0.04751 \\ \hline 0.004515 & -0.0002691 & 0 & 0 & 4.232e-5 \end{array}\right]$$

$$G_{34} \overset{s}{=} \left[\begin{array}{cc|ccc} 0.9873 & 0.1497 & 0.1109 & 24.72 & 0.04251 \\ -0.03991 & 0.9499 & -0.0585 & -1.138 & 0.08498 \\ \hline 0.006792 & -0.0005149 & 0 & 0 & 0.0001468 \end{array}\right]$$

$$G_{35} \overset{s}{=} \left[\begin{array}{ccc|ccc} 0.9958 & -0.009323 & 0.003507 & 0.5077 & 1.551e-11 & 0.0002144 \\ 0.0145 & 0.9915 & 0.1952 & 11.64 & -2.125e-12 & -0.007908 \\ -0.03425 & -0.01472 & 0.6606 & -19.29 & -2.313e-14 & -0.006491 \\ \hline 0.2266 & -0.0009256 & -0.0004529 & 0 & 0 & 7.062e-5 \end{array}\right]$$

14. Appendix B: Feedback Controllers Designs

$$K_1 \stackrel{s}{=} \left[\begin{array}{cc|c} 1.97 & -0.9704 & 4 \\ 1 & 0 & 0 \\ \hline -2.153 & 2.154 & 820 \end{array}\right]$$

$$K_2 \stackrel{s}{=} \left[\begin{array}{cc|c} 1.997 & -0.997 & 0.125 \\ 1 & 0 & 0 \\ \hline 0.07962 & -0.07684 & 40 \end{array}\right]$$

$$K_3 \stackrel{s}{=} \left[\begin{array}{ccc|c} 2.632 & -1.133 & 0.6343 & 16 \\ 2 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ \hline -5.395 & 5.381 & -5.367 & 240 \end{array}\right]$$

$$K_4 \stackrel{s}{=} \left[\begin{array}{cc|c} 1.638 & -0.6376 & 8 \\ 1 & 0 & 0 \\ \hline -4.491 & 4.491 & 100 \end{array}\right]$$

$$K_5 \stackrel{s}{=} \left[\begin{array}{cc|c} 1.942 & -0.9418 & 4 \\ 1 & 0 & 0 \\ \hline -1.085 & 1.085 & 100 \end{array}\right]$$

$$K_6 \stackrel{s}{=} \left[\begin{array}{c|c} 1 & 0.03919 \\ \hline 0.07655 & 125 \end{array}\right]$$

$$K_7 \stackrel{s}{=} \left[\begin{array}{ccc|c} 0.8941 & 0.001192 & -0.03769 & 2.278 \\ 0 & 0.9704 & 1 & 0 \\ 0 & 0 & 1 & 1.06 \\ \hline -1.487 & 0.02187 & -0.6915 & 41.78 \end{array}\right]$$

$$K_8 \stackrel{s}{=} \left[\begin{array}{cc|c} 1.97 & -0.09704 & 4 \\ 1 & 0 & 0 \\ \hline -2.386 & 2.391 & 400 \end{array}\right]$$

$$K_9 \stackrel{s}{=} \left[\begin{array}{cccc|c} 0.8435 & -0.03738 & -0.03978 & 0.3814 & 0 \\ 0 & 0.9851 & -0.01022 & 0.09802 & 0 \\ 0 & 0 & 0.9851 & 0.1043 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ \hline -0.3367 & -0.08654 & -0.09208 & 0.8829 & 0 \end{array}\right]$$

$$K_{10} \overset{s}{=} \left[\begin{array}{cc|c} .9246 & 0.0149 & 8.48 \\ 0 & 1 & 1.795 \\ \hline -5.767 & 1.221 & 696.4 \end{array}\right]$$

$$K_{11} \overset{s}{=} \left[\begin{array}{cc|c} 1.93 & -0.9301 & 2 \\ 1 & 0 & 0 \\ \hline 0.8487 & -0.8382 & 0 \end{array}\right]$$

$$K_{12} \overset{s}{=} \left[\begin{array}{cc|c} 1.93 & -0.9301 & 2 \\ 1 & 0 & 0 \\ \hline 1.697 & -1.676 & 0 \end{array}\right]$$

$$K_{13} \overset{s}{=} \left[\begin{array}{cc|c} 0.9704 & 1 & 0 \\ 0 & 1 & 0.1439 \\ \hline -0.005114 & 0.1439 & -1 \end{array}\right]$$

$$K_{14} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.4382 \\ \hline 0.2739 & 40 \end{array}\right]$$

$$K_{15} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.4382 \\ 0.2739 & 40 \end{array}\right]$$

$$K_{16} \overset{s}{=} \left[\begin{array}{cc|c} 0.9897 & 0.006134 & 0.1043 \\ 0 & 1 & 0.05883 \\ \hline -0.08812 & -0.04971 & -0.8451 \end{array}\right]$$

$$K_{17} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.4596 \\ \hline 0.6528 & 90.91 \end{array}\right]$$

$$K_{18} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.6197 \\ \hline 0.4841 & 50 \end{array}\right]$$

$$K_{19} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.445 \\ \hline 0.6742 & 6.061 \end{array}\right]$$

$$K_{20} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.4382 \\ \hline 0.3423 & 50 \end{array}\right]$$

$$K_{21} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.3098 \\ \hline 0.4841 & 24 \end{array}\right]$$

$$K_{22} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.1449 \\ \hline 0.207 & 1.429 \end{array}\right]$$

$$K_{23} \overset{s}{=} \left[\begin{array}{cc|c} 0.9494 & 0.01769 & 5.909 \\ 0 & 1 & 3.066 \\ \hline -8.655 & 4.492 & 1500 \end{array}\right]$$

$$K_{24} \overset{s}{=} \left[\begin{array}{cc|c} 0.9704 & 0.08849 & 0 \\ 0 & 1 & 0.125 \\ \hline -0.01713 & -0.1936 & 0 \end{array}\right]$$

$$K_{25} \overset{s}{=} \left[\begin{array}{cc|c} 0.9763 & 0.002556 & 0.7544 \\ 0 & 1 & 0.05422 \\ \hline -0.44 & -0.03162 & -9.332 \end{array}\right]$$

$$K_{26} \overset{s}{=} \left[\begin{array}{c|c} 1 & 0.36 \\ \hline 0.3333 & 14.81 \end{array}\right]$$

$$K_{27} \overset{s}{=} \left[\begin{array}{c|c} 1 & 22.17 \\ \hline 16.24 & 1.2e4 \end{array}\right]$$

$$K_{28} \overset{s}{=} \left[\begin{array}{cc|c} 1.05 & -0.1991 & 64 \\ 0.25 & 0 & 0 \\ \hline -10.35 & 41.41 & 700 \end{array}\right]$$

$$K_{29} \overset{s}{=} \left[\begin{array}{ccc|c} 1.1 & -0.2041 & 0.07932 & 32 \\ 0.5 & 0 & 0 & 0 \\ 0 & 0.0625 & 0 & 0 \\ \hline 0.5197 & -1.958 & 14.72 & 0 \end{array}\right]$$

$$K_{30} \overset{s}{=} \left[\begin{array}{cc|c} 1.997 & -0.997 & 16 \\ 1 & 0 & 0 \\ \hline 4.971 & -4.874 & 8000 \end{array}\right]$$

$$K_{31} \overset{s}{=} \left[\begin{array}{cc|c} 1.741 & -0.7408 & 32 \\ 1 & 0 & 0 \\ \hline -30.41 & 30.42 & 4000 \end{array}\right]$$

$$K_{32} \overset{s}{=} \left[\begin{array}{ccc|c} 3 & -1.5 & 0.4999 & 0.01563 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0.008383 & -0.008309 & 0.004117 & 0.01393 \end{array}\right]$$

$$K_{33} \overset{s}{=} \left[\begin{array}{ccc|c} 2.049 & -1.099 & 0.199 & 128 \\ 1 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ \hline 15.45 & -30.9 & 61.81 & -2089 \end{array}\right]$$

$$K_{34} \overset{s}{=} \left[\begin{array}{cc|c} 1.05 & -0.1991 & 64 \\ 0.25 & 0 & 0 \\ \hline 8.722 & -34.97 & -600 \end{array}\right]$$

15. Appendix C: Head Level Time Histories for Passive Thresholding Strategy

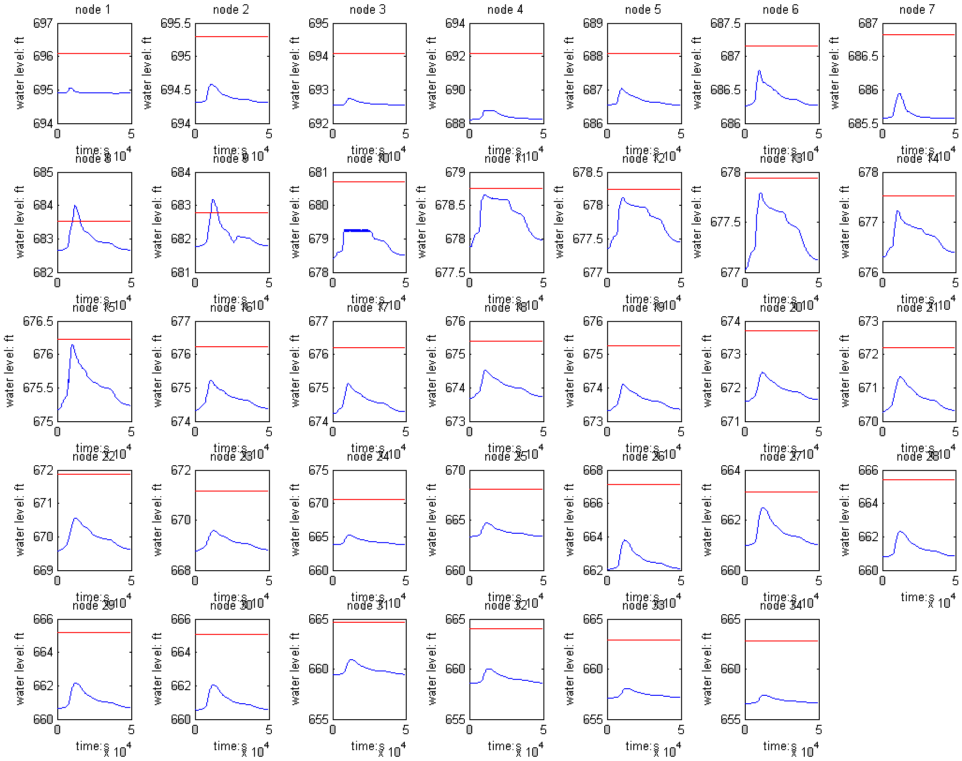


Figure 19. Storm C - Passive Threshold - South Bend Interceptor Sewer

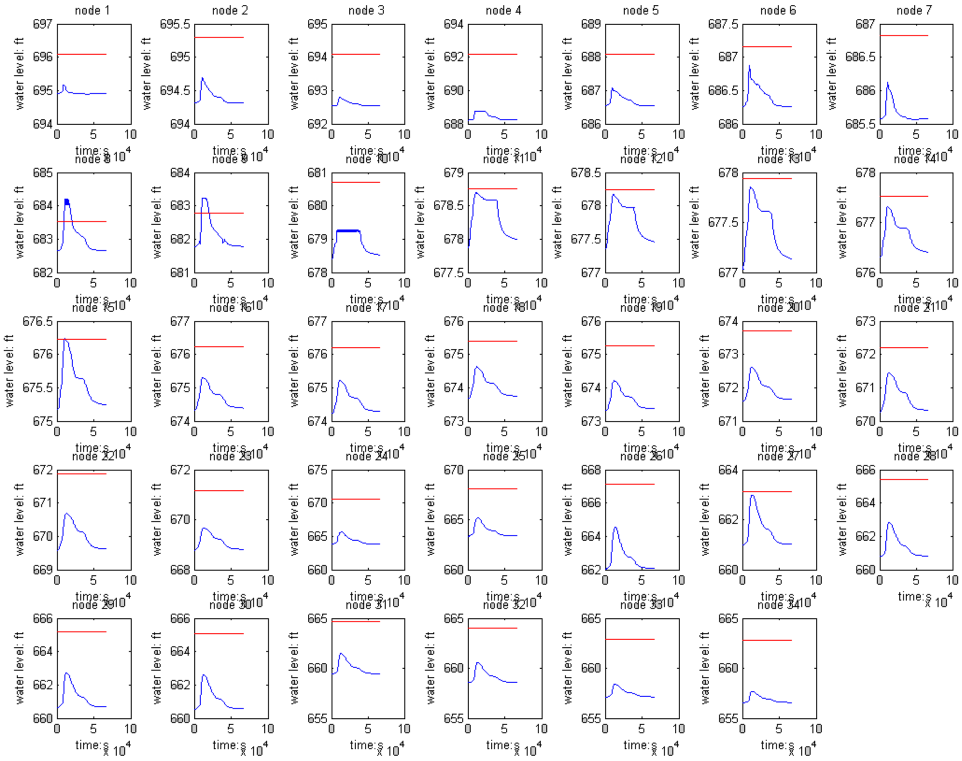


Figure 20. Storm D - Passive Threshold - South Bend Interceptor Sewer

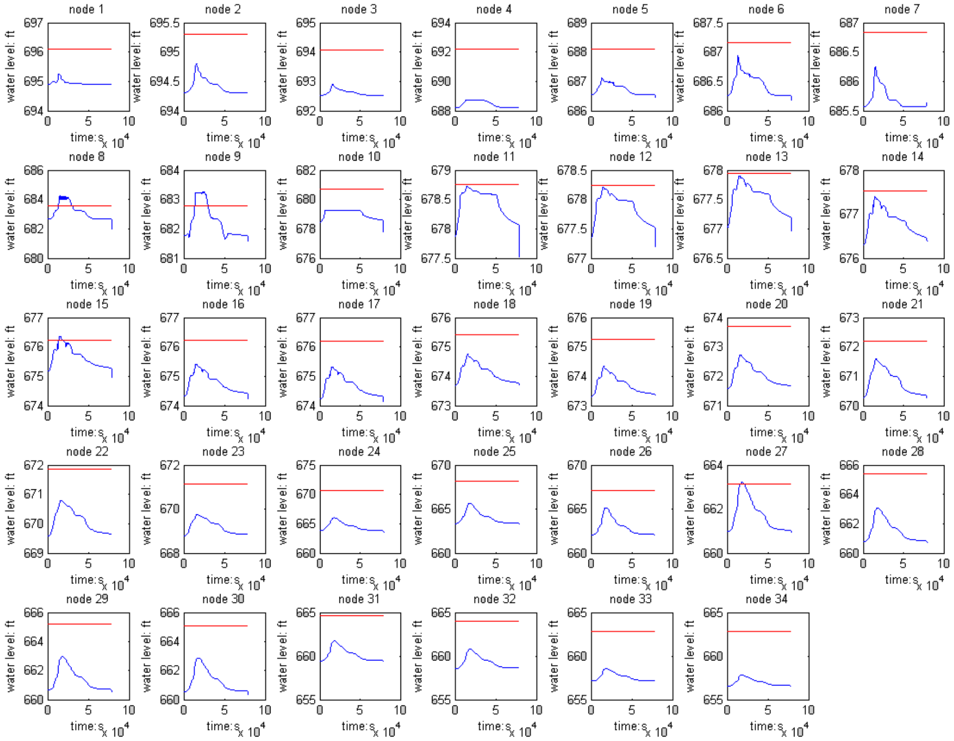


Figure 21. Storm E - Passive Threshold - South Bend Interceptor Sewer

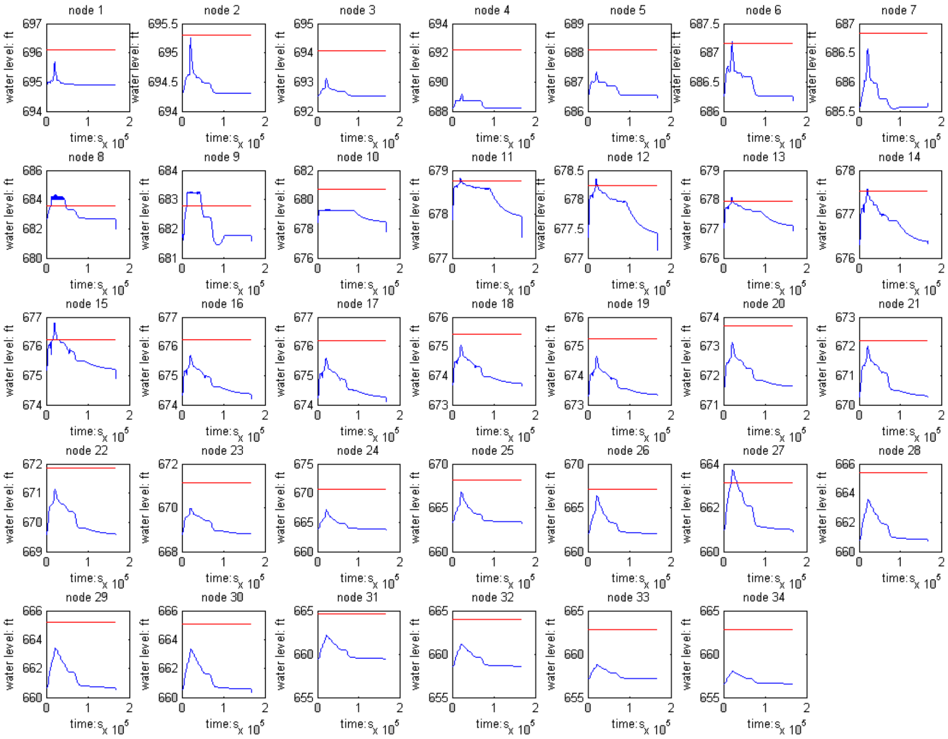


Figure 22. Storm G - Passive Threshold - South Bend Interceptor Sewer

16. Appendix D: Head Level Time Histories for Decentralized Control Strategy

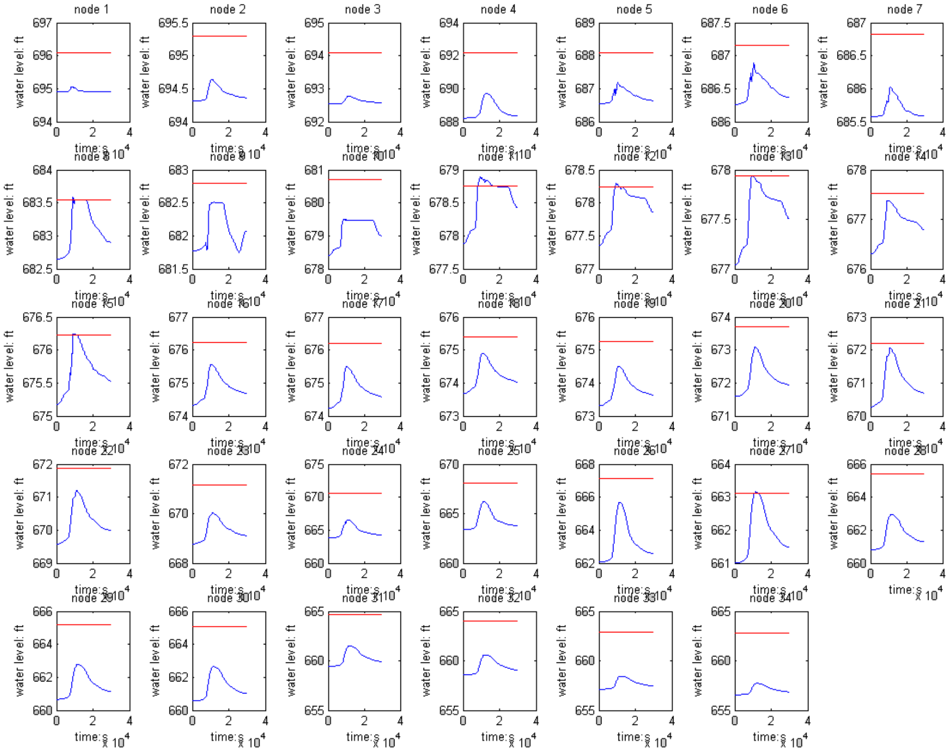


FIGURE 23. Storm C - Decentralized Control - South Bend Interceptor Sewer

REFERENCES

[1] M. Schutze, A. Campisano, H. Colas, W. Schilling, and P. A. Vanrolleghem, "Real time control of urban wastewater systems; where do we stand today?" *Journal of Hydrology*, vol. 299, pp. 335–348, 2004.

[2] M. Marinaki and M. Papageorgiou, "A non-linear optimal control approach to central sewer network flow control," *International Journal of Control*, vol. 72, no. 5, pp. 418–429, 1999.

[3] M. Marinaki, M. Papageorgiou, and A. Messmer, "Multivariable Regulator Approach to Sewer Network Flow Control," *Journal of Environmental Engineering*, vol. 125, no. 3, pp. 267–276, 1999.

[4] S. Duchesne, A. Mailhot, and J. Villeneuve, "Global Predictive Real-Time Control of Sewers Allowing Surcharged Flows," *Journal of Environmental Engineering*, vol. 130, no. 5, pp. 526–534, 2004.

[5] G. Cembrano, J. Quevedo, M. Salamero, V. Puig, J. Figueras, and J. Marti, "Optimal control of urban drainage systems. A case study," *Control Engineering Practice*, vol. 12, no. 1, pp. 1–9, 2004.

[6] T. Ruggaber, J. Talley, and L. Montestruque, "Using embedded sensor networks to monitor, control, and reduce cso events: A pilot study," *Environmental Engineering Science*, vol. 24, no. 2, pp. 172–182, 2007.

[7] P. Wan and M. Lemmon, "Distributed flow control using embedded sensor-actuator networks for the reduction of combined sewer overflow (cso) events," in *Proceedings of the IEEE Conference on Decision and Control*, 2007.

[8] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[9] A. A. et al., "Exscal: elements of an extreme scale wireless sensor network," in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2005.

[10] M. Chaudhry, *Open Channel Flow*. Prentice-Hall, 1993.

[11] E. P. Agency, "Storm water management model (swmm),"
http://www.epa.gov, March, 2007.

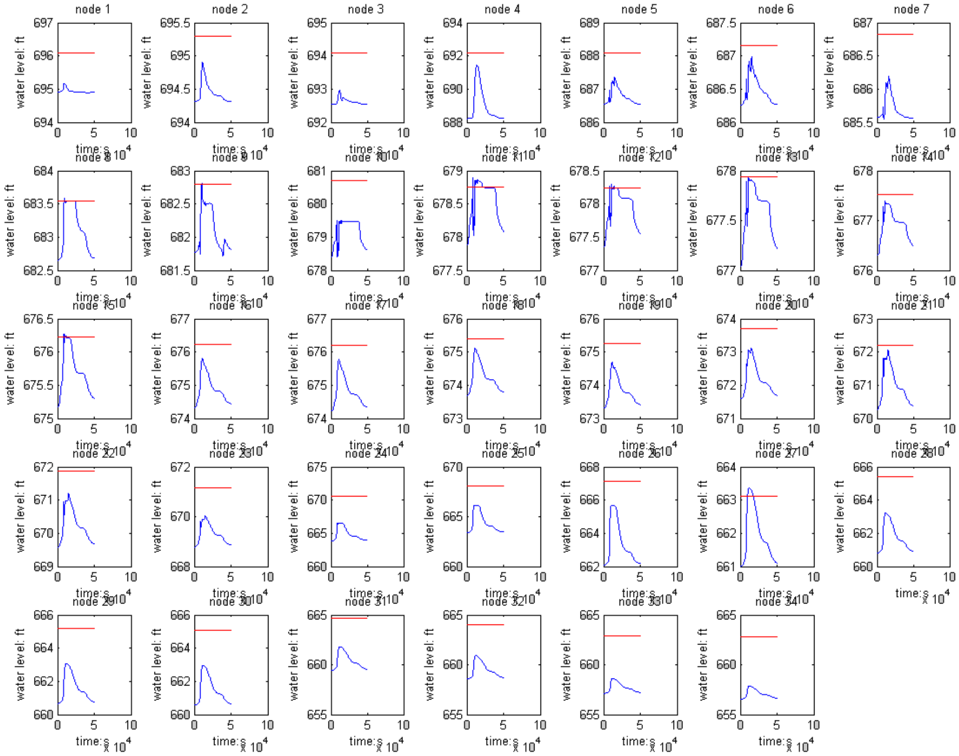


FIGURE 24. Storm D - Decentralized Control - South Bend Interceptor Sewer

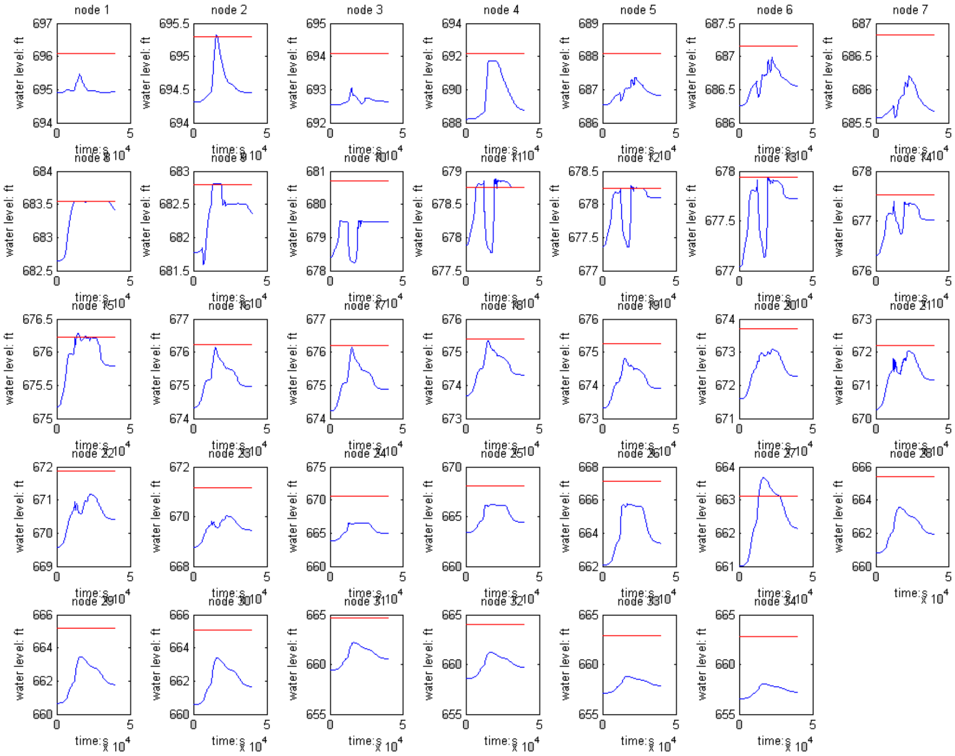


FIGURE 25. Storm E - Decentralized Control - South Bend Interceptor Sewer

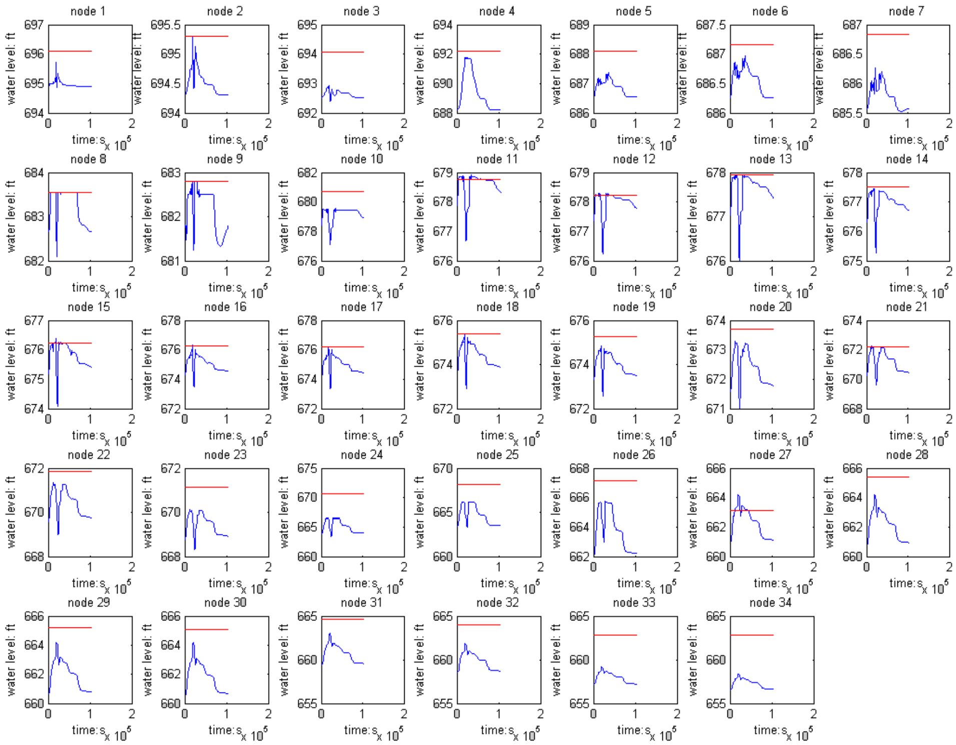


Figure 26. Storm G - Decentralized Control - South Bend Interceptor Sewer