*Water Resources Research*

Supporting Information for

# *Real-time control of urban headwater catchments through linear feedback: performance, analysis and site selection*

B. P. Wong[1], B. Kerkez[1]

[1]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor Michigan, USA

## Contents of this file



In Text S1, we present a condensed formulation of Section 3. In Text S2, we describe how to construct and implement a control model. In Text S3, a mathematical argument is offered which shows that the controller is robust to measurement noise. S3 presents a mathematical illustration that shows i) if one can measure all necessary states, the controller is robust to white noise, and ii) if there are states that cannot be measured (e.g. inaccessibility, broken sensor, etc.), using the separation principle, the design of the controller stays the same as presented in Section 3. Finally, in Text S4, we present additional results from the simulations using a long-term rainfall record. To test robustness, varying degrees of measurement noise (standard deviation, σ = 2.5, 12.5, and 25 mm) were also added to each simulation.

### Text S1.

The formulation of a linear quadratic regulator (LQR) controller requires a discrete time *linear* system with dynamics described by:

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

and a *quadratic* cost function parameterized by $\mathbf{Q}$ and $\mathbf{R}$

$$J = x^T(N)\,\mathbf{Q}\,x(N) + \sum_{k=0}^{N-1}\left(x^T(k)\,\mathbf{Q}\,x(k) + u^T(k)\,\mathbf{R}\,u(k)\right)$$

Then control input that minimizes the cost $J$ is given by

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

where the gain matrix $\mathbf{K}(k)$ is given by

$$\mathbf{K}(k) = (\mathbf{B_u^T}(k)\,\mathbf{P}(k)\;\mathbf{B_u}(k) + \mathbf{R})^{-1}(\mathbf{B_u^T}(k)\,\mathbf{P}(k)\;\mathbf{A}(k))$$

and $\mathbf{P}(k)$ is the solution to the discrete time Ricatti equation:

$$\mathbf{P}(k-1) = \mathbf{A^T}(k)\,\mathbf{P}(k)\,\mathbf{A}(k) - \left(\mathbf{A^T}(k)\,\mathbf{P}\,\mathbf{B_U^T}(k)\right)(\mathbf{B_u^T}(k)\,\mathbf{P}(k)\,\mathbf{B_u}(k) + \mathbf{R})^{-1}(\mathbf{B_u^T}(k)\,\mathbf{P}(k)\,\mathbf{A}(k)) + \mathbf{Q}$$

Note that neither $\mathbf{B_d}$ nor $d(k)$ appear in the cost function or the formulation of the gain matrix. In this study, $d(k)$ represents rainfall disturbances that the controller cannot account for. Intuitively, the control input $u(k)$ can only directly adjust the flows and water levels in the state vector $x(k)$ but has no effect on controlling the rainfall $d(k)$.

The formulation of the dynamics as discrete time linear system in this study uses an integrator delay model parameterized by physical properties.

The Integrator, used for each storage node:

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

The Delay, used for each conduit:

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

And the Link component, which connects the storage nodes and conduits:

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

The state vector $\boldsymbol{x}(\boldsymbol{k})$, control vector $\boldsymbol{u}(\boldsymbol{k})$, rainfall disturbance $\boldsymbol{d}(\boldsymbol{k})$, and matrices $\mathbf{A}(\boldsymbol{k})$, $\mathbf{B_u}(\boldsymbol{k})$, and $\mathbf{B_d}$, are then constructed by block-wise assembly of the Integrator, Delay, and Link components, as shown in (8), (9), (10) of the manuscript.

**Text S2.**

In Text S2, we describe how to construct and implement a control model parameterized by physical of the study catchment. The first step in the simulation process involves the abstraction of the physical watershed into a linearized control model. While this can be achieved manually, on a case-by-case basis, our approach automates this by first extracting physical parameters from a SWMM model and then converting them to the state-space formulation. Constructing a state-space representation of a SWMM model

begins with importing the properties of the SWMM model, including storage curves, contributing subcatchments, and the connectivity between links (pipes, channels, etc.) and storage nodes. These properties are then used to build the state, control, and disturbance matrices $\mathbf{A}(k)$, $\mathbf{B_u}(k)$, and $\mathbf{B_d}$.

The construction of these matrices is detailed in Algorithm 1 and Algorithm 2.

**Algorithm 1:** Construct/Update the state matrix, **A**, using properties from a SWMM model.

```
1:  Initialize:
    storages, junctions, conduits ← load_swmm_model()
    T ← simulation timestep
    A_s ← storage node surface area given current depth
    n ← 1

2:  for each node in storages do
3:      A(n,n) = A_integrator_matrix(T, A_s)
4:      n ← n+1
5:  end for
6:  for each node in junctions do
7:      A(n,n) = A_delay_matrix(T, A_s)
8:      n ← n+1
9:  end for
10: for each link in conduits
11:     m,n ← indices_of( link.head_node, link.tail_node )
12:     if head_node(link) is in storages then
13:         A(m,n) = A_link_matrix( lambda(k) )
14:     else if head_node(link) is in junctions then
15:         A(m,n) = A_link_matrix(1)
16:     end if
17: end for
```

**Algorithm 2:** Construct/Update the control matrix, $\mathbf{B}_u$, and disturbance matrix, $\mathbf{B}_d$.

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

2:  for each node in storages do
3:      Bu(n) = integrator_control_matrix(T, A_s)
4:      Bd(n) = integrator_disturbance_matrix(T, A_s)
5:      n ← n+1
6:  end for
7:
8:  for each node in junctions do
9:      Bu(n) = delay_control_matrix(1)
10:     Bd(n) = 0 * delay_control_matrix(1)
11:     n ← n+1
12: end for
```

If the physically-based model is not controlled, it can be executed in a stepwise fashion in Matlab or Python using Algorithm 3. The states of the model (water levels, flows, etc.) can then be extracted or visualized and the model can be halted once a specific state has been reached or total duration has been exceeded.

**Algorithm 3:** Execute a stepwise SWMM model simulation.

```
1:  while simulation is not over do
2:      swmm.step_forward
3:      k ← k+1
4:  end while
```

For the controlled case, the model is halted every step, after which the control matrixes of the linear model are updated (Algorithm 4). The outflow for each storage node is given by the control gain computed via an LQR control function and clipped, if necessary. The valve and gate positions are then set as a relative percentage of the total area. The loop is then repeated until the simulation period expires or specific state is reached.

**Algorithm 4:** Stepwise SWMM model simulation with LQR control.

```
1:  Initialize:
    get_swmm_model_properties()

2:  while simulation is not over do
3:          swmm.step_forward
4:          xhat[k] ← T * swmm.get_states()
5:          Ahat, Bhat, Qhat, Rhat ← update_system_matrices(xhat[k])
6:          Khat ← dlqr(Ahat, Bhat, Qhat, Rhat )
7:          Qoutflow ← Khat * xhat[k]
8:          Qclipped ← clip( Qoutflow, 0, Qmax )
9:          PercentOpen ← clip( Agate(Qclipped)/ Amax, 0, 1) * 100
10:         swmm.update_gate_positions(PercentOpen)
11:         k ← k+1
12: end while
```

**Text S3.**

In this study, we assume perfect, complete knowledge of the states of our system dynamics to derive a linear-quadratic controller and determine how all valves should be operated in coordination. However, this may not always be the case due to measurement noise or an inability to measure all the controllable states of our system. While filters such as moving averages or kernel smoothers may be useful in practice, noisy measurements, or lack thereof, can be remedied optimally with the use of a linear-quadratic estimator (i.e., Kalman filter). Below, we discuss how when assuming i) a noisy system and ii) a noisy system with missing measurements, the design of the feedback controller remains the same.

Suppose we have a noisy system with perfect measurement of the state

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

$$y(k) = \mathbf{C}\,x(k)$$

where

$$w(k) \sim \mathcal{N}(0,\, \mathbf{W}),\, \mathbf{W} > 0,$$

$$x(k) \sim \mathcal{N}(0,\, X_0),\, X_0 = \mathcal{E}[x(t_0)\, x^T(t_0)]$$

Given the noise in our system, we modify our cost function to take the average over all possible realizations using the expected value:

LQR: Updated Cost Function

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

where

$$\mathbf{Q},\, \mathbf{R} > 0$$

Since $w(k)$ is not a term that can be reduced by $\mathrm{u}(k)$ in the cost function and $w(k)$ is white, $\mathcal{E}\{w(k)\} = 0$, the optimal controller for this case is identical to the deterministic case.

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

Illustrating the Separation Principle

Suppose we want to design a controller for a system with both incomplete and noisy measurements

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

$$y(k) = \mathbf{C}\,x(k) + v$$

where

$w(k) \sim \mathcal{N}(0,\, \mathbf{W})$, with covariance $\mathbf{W} > 0$,

$v(k) \sim \mathcal{N}(0,\, \mathbf{V})$, with covariance $\mathbf{V} > 0$,

$x(k) \sim \mathcal{N}(0,\, X_0),\, X_0 = \mathcal{E}[\{x(0) - \mathcal{E}[x(0)]\}\{x(0) - \mathcal{E}[x(0)]\}^T]$

and our Cost Function is the Updated Cost Function:

$$J = \mathcal{E}\left[x^T(N)\,\mathbf{Q}\,x(N) + \sum_{k=0}^{N-1}(\rho \cdot x^T(k)\,\mathbf{Q}\,x(k) + u^T(k)\,\mathbf{R}\,u(k))\right]$$

By the Separation Principle, the control input that minimizes $J$ is

$$u(k) = -\mathbf{K}\,\hat{x}(\mathrm{k})$$

where

$\hat{x}\,(k) =$ the optimal estimate from $y(k)$ using gain $\mathbf{L}$ that minimizes

$$\mathcal{E}\left[\{x(k) - \hat{x}\,(k)\}\{x(k) - \hat{x}\,(k)\}^{T}\right]$$

$$\mathbf{K} = (\mathbf{B_u^T}(k)\,\mathbf{P}(k)\,\mathbf{B_u}(k) + \mathbf{R})^{-1}(\mathbf{B_u^T}(k)\,\mathbf{P}(k)\,\mathbf{A}(k))$$

$$\mathbf{L} = (\mathbf{S}(k)\,\mathbf{C^T})(\mathbf{C}\,\mathbf{S}(k)\,\mathbf{C^T} + \mathbf{V})^{-1}$$

The Separation Principle guarantees $\mathbf{K}$ and $\mathbf{L}$ can be computed independently, where $\mathbf{K}$ is the control gain from a linear-quadratic regulator and $\mathbf{L}$ is the estimator gain from a linear-quadratic estimator (i.e., Kalman filter). $\mathbf{P}(k)$ is the solution to the discrete time *Ricatti* equation and $\mathbf{V}$ is the covariance of the measurement noise $v(k)$. This solution is known as Linear Quadratic Gaussian Control.

The system dynamics for the estimate $\hat{x}\,(k)$ are given by:

$$\hat{x}\,(k+1 \mid k) = \mathbf{A}(k)\;\hat{x}\,(k \mid k-1) + \mathbf{B_u}(k)\,u(k) + \mathbf{L}\left[y(k) - \mathbf{C}\;\hat{x}\,(k \mid k-1)\right]$$

The term $y(k) - \mathbf{C}\;\hat{x}\,(k \mid k-1)$ is known as the *Innovation* – the difference between the observed value $y(k)$ at timestep k and the optimal forecast based on prior information. If $\hat{x}\,(k)$ is indeed the optimal estimate, successive estimates are uncorrelated with each other and the innovation

$$i(k) \equiv y(k) - \mathbf{C}\;\hat{x}\,(k \mid k-1),\; i(k) \sim \mathcal{N}(0,\; \mathbf{V} + \mathbf{C}\,\mathbf{S}(k)\,\mathbf{C^T})$$

can be defined as white noise (Gelb et al. 1974).

Then $\hat{x}\,(k+1 \mid k)$ can be rewritten as

$$\hat{x}\,(k+1 \mid k) = \mathbf{A}(k)\;\hat{x}\,(k \mid k-1) + \mathbf{B_u}(k)\,u(k) + \mathbf{L}\,i(k)$$

Minimizing the updated cost function is independent of the innovation noise and the control input remains the same as before for both the estimated and noisy, incomplete systems:

$$u(k) = -\mathbf{K}\;\hat{x}(\mathrm{k})$$

In this study, we focus exclusively on a framework for developing system level controller and assume $\hat{x}\,(k) = x(k)$ to arrive at the controller input derived in the Methods section.

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

We hope more sophisticated approaches will be considered under this framework in future studies, such as setpoint and predictive control.

**Text S4.**

The SWMM model in our study was calibrated using events from 2013, spanning from April 1 to December 1, 2013. The rainfall record in this study and was collected from Weather Underground. Invalid measurements (reported value of -99.99) were ignored in the simulation. This weather station (KMIANNAR41) was the closest one to our study catchment that had a 2013 rainfall record, which was reported at approximately five-minute intervals. The uncontrolled case was compared against the controlled case with four (Nodes 4, 6, 10, 11) and eleven control points, as well as four control points at 50% storage volume, each with varying degrees of noise (σ = 2.5, 12.5, and 25 mm).

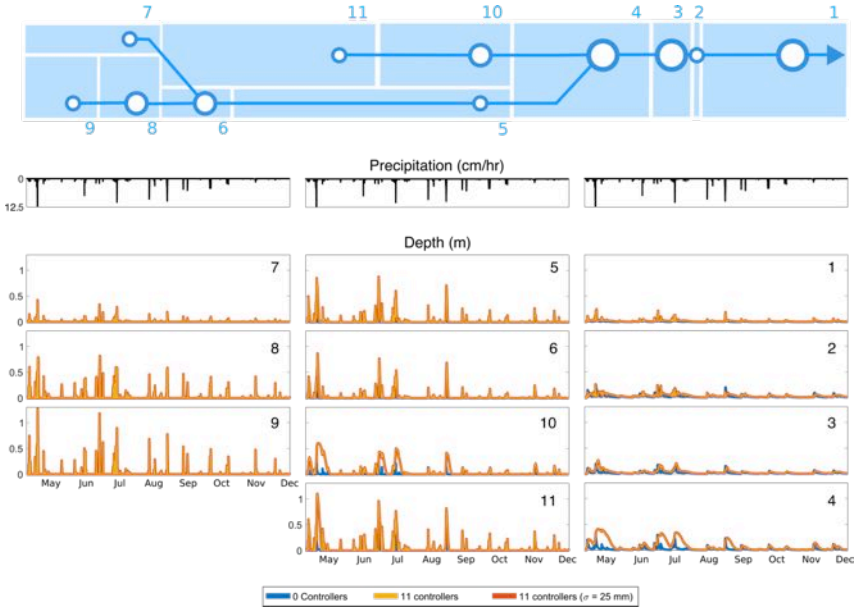


**Figure S1.** Rainfall and storage depths for April 1 to December 1, 2013 for both the uncontrolled and controlled cases with eleven controllers, simulated with perfect and noisy (σ = 25 mm) measurements.

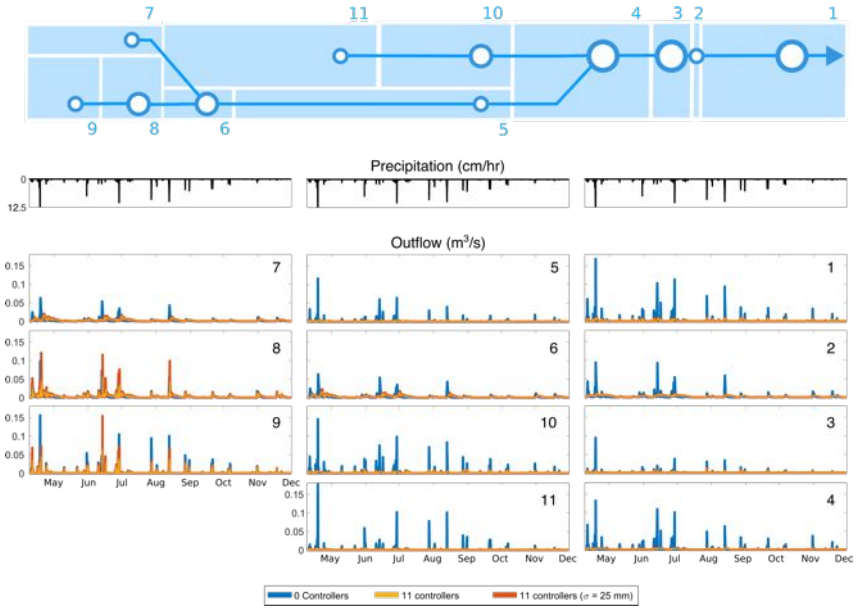


**Figure S2.** Rainfall and outflows depths for April 1 to December 1, 2013 for both the uncontrolled and controlled cases with eleven controllers, simulated with perfect and noisy (σ = 25 mm) measurements.

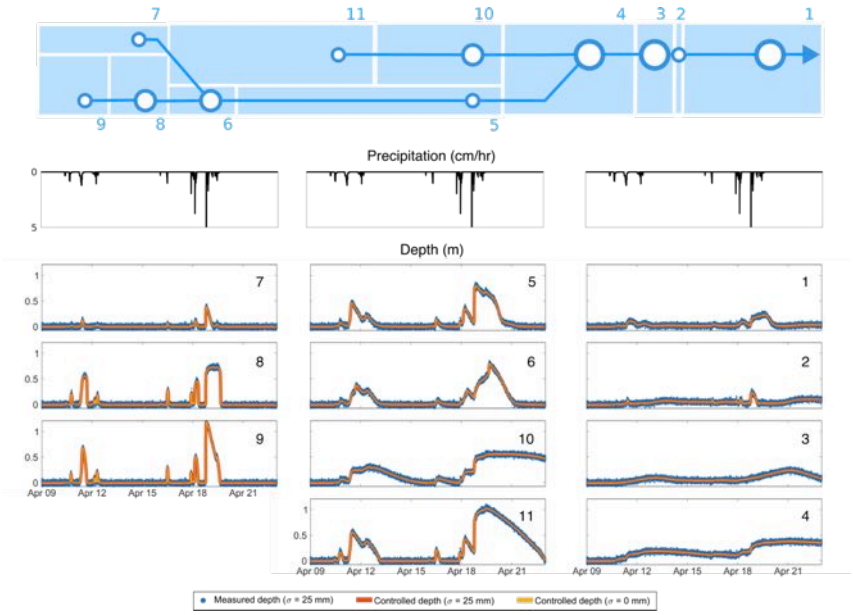


**Figure S3.** Measured and controlled depths for (σ = 0 and 25 mm) for April 9-23, 2013. The response to rainfall remains similar in both cases despite the additional noise.

| Long-term performance summary ($P_{mod}$) | | | | | |
|---|---|---|---|---|---|
| | No Control | Control + measurement noise σ | | | |
| | | σ = 0 mm | σ = 2.5 mm | σ = 12.5 mm | σ = 25 mm |
| 11 ponds | 1236.57 | 69.85 | 69.86 | 70.87 | 76.87 |
| 4 ponds | 1236.57 | 227.40 | 227.34 | 227.42 | 227.30 |
| 4 ponds @ 50% | 1767.85 | 359.81 | 359.80 | 359.87 | 359.93 |

**Table S1.** Long-term system performance with $\boldsymbol{Q^*_{i,max}}$ = 0.029 $m^3/s$ using 2013 rainfall data. Nodes 4, 6, 10, and 11 are controlled in the 4-pond case and also modified to half their storage in the 50% case. Performance is improved in all control cases compared to the passive system.

**Storage Node Properties**

| Node | Volume ($m^3$) | Average Area ($m^2$) | Max Height (m) | Catchment Area ($m^2$) | % Impervious |
|---|---|---|---|---|---|
| 1 | 31 700 | 10 300 | 3 | 147 000 | 87 |
| 2 | 800 | 200 | 3 | 5 400 | 74 |
| 3 | 30 900 | 10 100 | 3 | 41 000 | 32 |
| 4 | 31 800 | 10 400 | 3 | 14 500 | 78 |
| 5 | 1 290 | 400 | 3 | 92 500 | 51 |
| 6 | 16 100 | 2 600 | 6 | 22 700 | 41 |
| 7 | 2 100 | 600 | 3 | 43 300 | 38 |
| 8 | 4 000 | 1 300 | 3 | 41 000 | 51 |
| 9 | 367 | 100 | 3 | 51 700 | 35 |
| 10 | 19 500 | 4 600 | 4 | 93 800 | 75 |
| 11 | 7 800 | 1 700 | 4 | 145 000 | 48 |

**Table S2.** Select physical properties of the storage nodes.

**Conduit Properties**

| Start Node | End Node | Shape | Max Height (m) | Length (m) | Manning N |
|---|---|---|---|---|---|
| 2 | 1 | Circular | 0.30 | 240 | 0.01 |
| 3 | 2 | Circular | 0.30 | 55 | 0.01 |
| 4 | 3 | Circular | 0.30 | 280 | 0.01 |
| 5 | 4 | Circular | 0.30 | 56 | 0.01 |
| 6 | 5 | Circular | 0.30 | 360 | 0.01 |
| 7 | 6 | Circular | 0.43 | 42 | 0.01 |
| 8 | 6 | Circular | 0.30 | 190 | 0.01 |
| 9 | 8 | Circular | 0.30 | 81 | 0.01 |
| 10 | 4 | Circular | 2.00 | 190 | 0.13 |
| 11 | 10 | Circular | 0.30 | 340 | 0.01 |

**Table S3.** Select physical properties of the conduits between storage nodes

## Supporting Information References

Gelb A, Kasper Jr J, Nash Jr R, Price C, Sutherland Jr A. 1974. Applied optimal estimation, A. gelb, ed. .