International Conference on Hydroinformatics

2014

Adaptive, Decentralized, And Real-Time Sampling Strategies For Resource Constrained Hydraulic And Hydrologic Sensor Networks

Brandon Preclaro Wong

Branko Kerkez

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

ADAPTIVE, DECENTRALIZED, AND REAL-TIME SAMPLING STRATEGIES FOR RESOURCE CONSTRAINED HYDRAULIC AND HYDROLOGIC SENSOR NETWORKS

BRANDON P. WONG (1), BRANKO KERKEZ (1)
(1): Civil Infrastructure Systems, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, Michigan, USA

We propose a low-power sensor node that can autonomously adjust the sampling interval of a suite of sensors based on local state estimates and future predictions of water flow. The problem is motivated by the need to accurately reconstruct abrupt state changes in urban watersheds and stormwater systems. Presently, the detection of these events is limited by the temporal resolution of sensor data. It is often infeasible, however, to increase measurement frequency due to energy and sampling constraints. We introduce a novel sampling algorithm, which queries local weather forecasts to anticipate state changes in a hydrograph signal. Initial results indicate that the algorithm effectively decouples the rising and receding limbs of the hydrograph, and thus has the potential to significantly reduce the power consumed by a field-deployed sensor node.

1. INTRODUCTION

Our problem is motivated by the need to accurately reconstruct abrupt state changes in urban watersheds and stormwater systems. These state changes often contain significant information that can be used to better study and manage water systems. For example, sudden rises in hydrographs due to precipitation events often drive the *first flush* of nutrients and other contaminants into streams and rivers [1]. Water quality studies thus benefit significantly from the ability to detect and measure these events. Presently, the detection of these events is limited by the temporal resolution of sensor data.

Increasing the sampling frequency of sensors is often infeasible, however. While advances in microcontroller and radio design have enabled the reliable and low-power collection of data, a major driver of power consumptions is the sensor itself [2]. Power consumption thus becomes a main factor when considering the ubiquitous, battery-powered deployment of hydrologic and hydraulic sensor networks. Furthermore, reliable and cost-effective sensors for many water quality parameters have yet to be developed. In these cases, it is often possible to use a sampling device instead, which can automate the collection of a small set of field samples for subsequent analysis in the laboratory. In such systems, the total number of available samples becomes a major constraint. Wasted samples are a major problem in these applications, where samples are taken even if the system being studied is not undergoing significant change. To alleviate this issue, it has been suggested that the sampler can be triggered when a flow sensor detects that the hydrographs has exceed a given flow threshold. The falling limb of the hydrograph can then be sampled evenly according to a predetermined sampling interval [3]. Such methods do not however anticipate precipitation events and often miss the onset of the ascending limb of the hydrograph, especially if the rising limb of the hydrograph does not match the predetermined threshold levels.

Given these resource constraints, we would thus like to develop a means by which to automate the measurement of hydrologic and hydraulic signals, which minimizes the number of required samples, while maximizing the measured amount of "*interesting*" information. We introduce such a method in this paper. Our algorithm optimizes the sampling frequency of a hydrologic sensor based on real-time weather forecasts, while being simultaneously simple enough to run on a real-world, low-power microcontroller. We will illustrate the efficacy of the proposed sampling algorithm on a real-world streamflow data set.

2. BACKGROUND

Adaptive sampling

The concept of adaptively changing a measurement strategy has been introduced in the signal processing and machine learning literature for a variety of applications. In most cases, the problems focus around a spatial phenomenon, where measurement at one location is used to guide the next sampling coordinate [4]. In the time domain, the objective is often based on the need to increase or decrease the sampling frequency based on events of interest. In many such cases the problem can be decoupled into:

- *Change point detection*: monitoring a signal to determine when an event of interest occurs [5, 6].
- *Adaptive sampling*: changing the sampling frequency to detect and measure events of interest [7].

This often presents a challenge, since both objectives are inherently intertwined.

The motivating example in this paper entails the real-time separation and prediction of the rising and falling limbs of a hydrographs (Figure 1). A rise in the hydrograph signal (Day 4, Figure 1) is the results of precipitation events, where, following a rain event, the flow of water increases for a relatively short duration of time. The rising limb of the hydrograph is of often of particular interest, since water quality measurements often contain the most amount of information in this region due to surface runoff. To that end, we would like to formalize the following requirement: *sample the rising limb at a high frequency, while sampling the remainder of the signal more slowly*. Furthermore, we would like the resulting technique to be computationally simple enough to execute efficiently on a field-deployable microcontroller.

While conserving energy, sampling too slowly can lead to missed data points. Sampling at a high frequency, on the other hand, will ensure the detection of the rising limb, but is not feasible due to resource constraints, especially if battery powered operations are a desired deployment outcome. Furthermore, high sampling frequencies often result in wasted samples, as the system being studied does not change fast enough (Figure 1 shows a significant amount of noise in the signal for a real-world hydrograph).

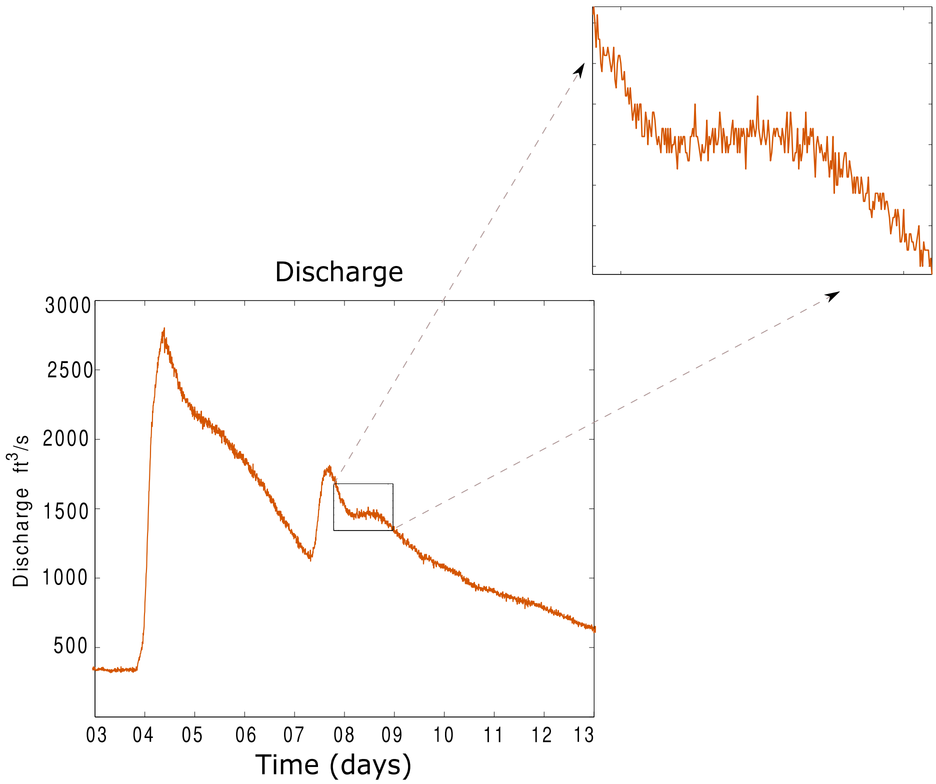

Figure 1. Example hydrograph obtained from US Geological Survey. Significant sensor noise is evident upon close inspection.

Hardware and system design

To enable the spatiotemporally dense measurement of water flow and water quality across urban watersheds we have developed and deployed a novel network of wireless sensor nodes. Each of our sensor nodes is equipped with a low-power microcontroller (Cortex-M3 architecture) and a low-power wireless module to take advantage of urban cellular coverage [2]. A prototype sensor network has recently been deployed in an urban watershed in Michigan and initial results indicate the system provides a robust and low-cost means by which to instrument large urban areas. IP-connectivity permits each node to query public web services for precipitation forecasts. Based on these forecasts, a node will persistently update a local, embedded model of flow conditions (described below) to adaptively change its sampling strategy to increase the likelihood of capturing abrupt changes in a sensor signal. This architecture will form an embedded processing chain, leveraging local computational resources to assess uncertainty by analyzing data as it is collected. The system is entirely decentralized and self-sustainable, requiring no server-side intervention or control. Each node only relies on a public weather forecast, which it can obtain via its Internet connection. The system is presently being assessed in an 800 square kilometer watershed near Ann Arbor, Michigan. Because the system was only recently deployed, we will evaluate the performance of the sampling algorithm by conducting a playback simulation on densely sampled, publically available streamflow data collected at a nearby site.

3. METHODS

Assume that we are interested in measuring a signal $x(t)$, which is corrupted by normally distributed, zero mean noise. At any point in time, a sensor measurement is given by:

$$y(t) = x(t) + \varepsilon \tag{1}$$

where $\varepsilon \sim N(0, \sigma^2)$. We make the assumption that the sensor noise is stationary and that the variance is known (through manufacturer datasheets, for example). Upon taking a sample at time t, the controller places the sensor reading in a sampling vector $y = \{y_1, y_2, \dots, y_N\}$ and a corresponding time stamp vector $t = \{t_1, t_2, \dots, t_N\}$. Following a reading at time t, our objective is to decide at which point in time to take a next reading, given by $t + t_s$. To minimize sensor power consumption, we would like to design our controller in a way that samples a signal quickly during events of interest, and slowly during less "*interesting*" periods. While there are many ways to encode this objective, in many water applications we are particularly interested in tracking two flow regimes: the rising limb of the hydrograph, and the receding limb of the hydrograph. These regimes can be detected by differentiating the sensor signal:

$$\begin{aligned} &\frac{dx}{dt} \geq 0 \quad \text{increasing hydrograph limb} \\ &\frac{dx}{dt} < 0 \quad \text{decreasing hydrograph limb.} \end{aligned} \tag{2}$$

Note that we require the derivative of the noise free signal. Given the high noise levels in real-world signals (see Figure 1), we cannot simply differentiate the measured signal y, as the result would only amplify the effects of the noise. To that end, we derive an estimate $\hat{x}(t)$ of noise free reading through a non-parametric kernel smoother [8]. For a noisy observation y_j at time t_j let $\hat{x}(t_j): \mathbb{R}^{\mathrm{N}} \rightarrow \mathbb{R}$ be a function that obtains a local estimate of x_j through the kernel average:

$$\hat{x}(t_j) = \hat{x}_j = \frac{\sum_{i=1}^{N} K(t_j, t_i) y_i}{\sum_{i=1}^{N} K(t_j, t_i)} \tag{3}$$

where K is the smoothing kernel. Given our normally distributed noise assumption, a good choice of kernel is given by the radial basis function:

$$K(t^*, t_i) = exp\left(-\frac{(t^* - t_i)^2}{2b^2}\right) \tag{4}$$

where b is the length-scale parameter. The kernel smoothing method can be thought of as a moving average filter, which weighs the importance of neighboring measurements based on their distance (time, in this case) to the measurement of interest. This smoothing method is also preferred because it does not require measurements to be sampled at even intervals. Once the measured data has been smoothed, an estimate of the noise free derivative can be obtained by numerically differentiating the smoothed data. Namely, we can assume that $\frac{dx}{dt} = \frac{d\hat{x}}{dt}$.

We now define a map, or sampling rule, $f(x): x \rightarrow t_s$, which determines how soon to take the next sample (t_s) based on a set of existing measurement estimates $\hat{x}$. For example, assume that we have a weather forecast (in the form of a precipitation probability $p_{rain} \in [0,1]$) we can design a controller that samples at a high frequency only if it detects a rising hydrograph or a chance of precipitation:

$$f(x) = t_s = \begin{cases} t_{short} & if \ \left(\frac{dx}{dt} \geq 0\right) \ \vee \ (p_{rain} > 0) \\ t_{long} & if \quad \text{otherwise} \end{cases} \tag{5}$$

Above, $\frac{dx}{dt}$ is evaluated using the latest pair of noise free sensor readings $\hat{x}$. We define the intervals such that $t_{long} > t_{short}$. If we are on the rising limb of the hydrograph (positive derivate), the controller takes the next sample at $t + t_{short}$ (higher sampling frequency). The converse is true for the falling limb. While simple, this rule is powerful: the controller is constantly updating its estimate of the noise free values $\hat{x}$ and using this estimate to change the sampling frequency in real-time. Furthermore, the weather forecast is used to anticipate a change in the hydrograph to ensure that a sudden rise in the hydrograph is not missed if it occurs. While beyond the scope of this paper, the above sampling rule can easily be modified to allow for more complex sampling strategies. For example, instead of sampling the decreasing limb of the hydrograph at a constant rate, one could use an exponential sampling rule to take more samples near the hydrograph peak and fewer samples away from the peak.

4. RESULTS

A playback analysis was conducted on a densely sampled streamflow data set (USGS 04165500 Clinton River[1]). This streamflow gage is substantially more complex than our proposed sensor nodes and provides readings at five-minute intervals. Historical precipitation forecast were obtained through the US National Weather Service[2], each consisting of six hour forecast windows. The sampling algorithm was implemented in MATLAB, with $t_{short} =$ 0.5 hours and $t_{long} = 3$ hours.

For brevity, we show results during April-May of 2012. The algorithm changed sampling strategies on a number of occasions between 4/28/2012 and 05/19/2012, driven primarily by precipitation forecasts. High sampling frequencies remained in effect until a fall in the hydrograph was detected. During two occasions, positive precipitation forecasts did not correspond with a rise in the hydrograph. A full reconstruction of the re-sampled hydrograph, as chosen by our algorithm, is shown in the bottom of Figure 2. The resampled time series retained both the rising and falling features of the highly-sampled hydrograph. For the dates indicated in Figure 2, if the original hydrograph had been sampled at intervals equal to $t_{short} = 0.5$ hours (fast sampling frequency), it would have required 230% more samples when compared to the adaptive sapling algorithm.

---

[1] http://waterdata.usgs.gov/usa/nwis/rt
[2] http://www.crh.noaa.gov/dtx/

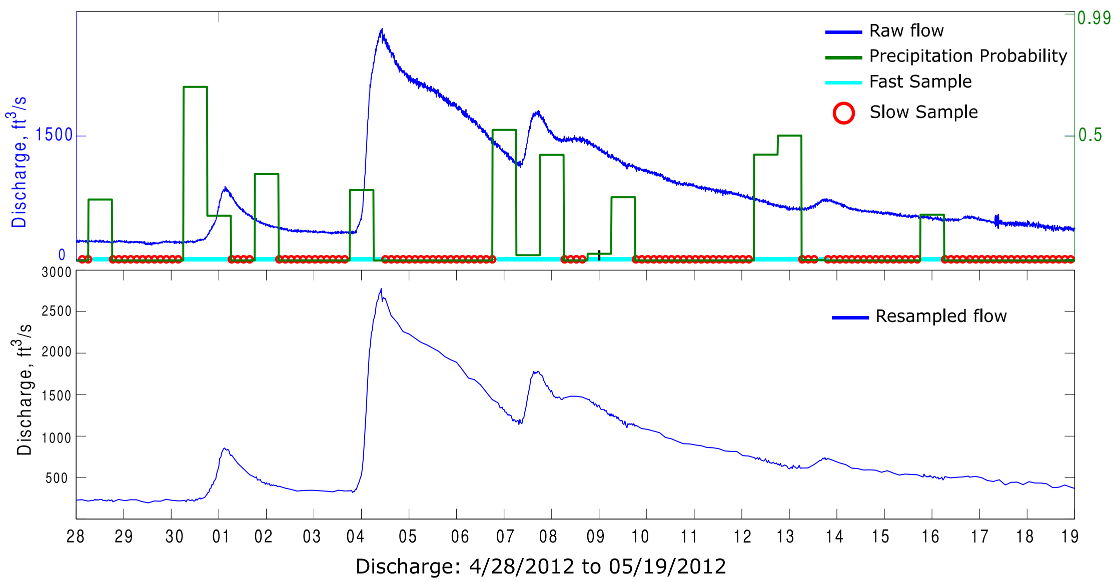

Figure 2. Sampling results. Original, over-sampled signal (top figure). Rising and falling limbs of the hydrograph sampled at different frequencies (bottom figure).

6. CONCLUSIONS

Initial results indicate that a simple adaptive sampling algorithm performs well in capturing the rinsing and falling limbs of a hydrographs. The hydrograph features were preserved, while the number of samples was significantly reduced compared to presently employed methods. While more rigorous analysis is warranted, this suggests that an Internet-enabled sensor node could significantly reduce the number of required samples (and thus its power consumption) by leveraging real-time weather forecast. Furthermore, these forecasts will ensure that the initial rise in the hydrograph is captured. The sampling rules used in this preliminary analysis will be expanded in the future to further reduce the amount of required samples during the hydrograph recession. As opposed to just flow, water quality will also be addressed in the context of this algorithm.

Acknowledgements

We would like to acknowledge the support of the University of Michigan, as well as our collaborators at the Huron River Watershed council for facilitating the deployment of our sensor nodes on the Huron River in Ann Arbor, Michigan.

REFERENCES

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

[2] Y. Zhao and B. Kerkez, "'Cellular-enabled water quality measurements," *AGU Fall Meeting 2013*, vol. 1.

[3] H.E. Gall, C.T. Jafvert and B. Jenkinson, "Integrating hydrograph modeling with real-time flow monitoring to generate hydrograph-specific sampling schemes," *Journal of Hydrology*, vol. 393, no. 3, pp. 331.

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

[5] Tze Leung Lai, "'Sequential changepoint detection in quality control and dynamical systems," *Journal of the Royal Statistical Society, 1995.*

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

[7] S. Thompson, A. George and S. Frederick, "*Adaptive sampling*", vol. 2014, ed. http://www.mathstat.helsinki.fi/msm/banocoss/2011/Presentations/Thompson_web.pdf, 1996.

[8] T. Hastie, R. Tibshirani and J. Friedman, "The Elements of Statistical Learning,".