

Leak detection in open water channels^{*}

Erik Weyer^{*} Georges Bastin^{**}

^{*} *Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia (e-mail: e.weyer@ee.unimelb.edu.au).*

^{**} *Center for Systems Engineering and Applied Mechanics (CESAME), University of Louvain-La-Neuve, Batiment Euler, 4 Avenue G. Lemaitre, B-1348 Louvain-la Neuve, Belgium (e-mail: bastin@inma.ucl.ac.be)*

Abstract: In this paper we present a simple cumulative sum algorithm for detection of leaks in open water channels. The algorithm compares the observed changes in water levels against the known in- and out-flows and raises an alarm if they are not in agreement. The algorithm is tested on data from an irrigation channel with very good results. Leaks are quickly detected and the algorithm is robust against uncertainty in the model parameters.

Keywords: Fault detection, CUSUM algorithm, system identification, leak detection, irrigation channels.

1. INTRODUCTION

Water is becoming an increasingly scarce resource in many parts of the world, and agriculture is one of the biggest consumers of water (Mareels et. al. 2005). Water for agricultural purposes is often conveyed via a network of open channels. Management of such channels must take into account the required level of service that has to be provided to the farmers and the need to conserve water and minimise potential water losses. Implementation of automatic control systems for regulation of the flows and water levels (Cantoni et. al. 2007, Dulhoste et. al. 2004, Gomez et al. 2002, de Halleux et. al. 2003, Litrico et. al. 2005, Ooi and Weyer 2007, Schuurmans et. al. 1999, Weyer 2007) can give significant improvement in operational efficiency, ensuring that enough water is available to farmers while minimising losses due to oversupply which can cause spillage along the channel and outflows at the end of the channel system.

In addition to losses caused by oversupply of water, there are also losses due to faults (Bedjaoui et. al. 2006) and leaks in the channels. Leaks can e.g. occur in the form of unscheduled offtakes of water or be due to break downs and failures in the (often old) civil engineering infrastructure. A typical example is a gate to an escape channel not sealing properly and letting water through even when it is fully closed. It is of course important to detect leaks early such that corrective action to reduce the water losses can be taken.

In this paper we present a simple algorithm for leak detection together with experimental results from the Coleambally Channel Number 6 which is a fully operational irrigation channel. The algorithm is a cumulative sum (CUSUM) algorithm which compares the observed changes

^{*} This work was supported by Rubicon Systems Australia and the Australian Research Council under the Linkage Grant Scheme, Project LP0349134.



Fig. 1. The Coleambally Channel Number 6

in water level against the measured and known in- and out-flows and raises an alarm if they are not in agreement.

The paper is organised as follows. In the next section we give a description of the Coleambally Channel no. 6 and present the models used. The leak detection algorithm is derived in Section 3 followed by experimental results in Section 4. Conclusions are given in Section 5.

2. CHANNEL DESCRIPTION

The experiments were carried out at the Coleambally Channel Number 6 (Fig. 1) which is a secondary channel to the Coleambally Main Channel in New South Wales, Australia. Fig. 2 shows a schematic top view of the channel.

We refer to a stretch between two gates as a pool. The pools are named according to the upstream gate, e.g. the pools in Fig. 2 are Pools 1 to 5. The channel is automated with overshoot gates as shown in Fig. 3 where y_i and y_{i+1} are the upstream water levels of Gates i and $i + 1$

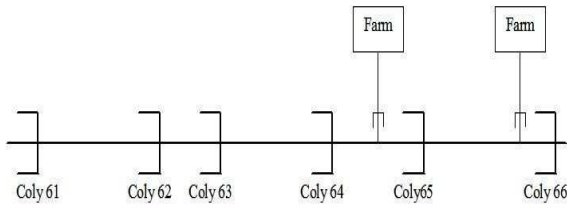


Fig. 2. Topview of the Coleambally Channel no. 6 with Gates 1 to 6 (not to scale).

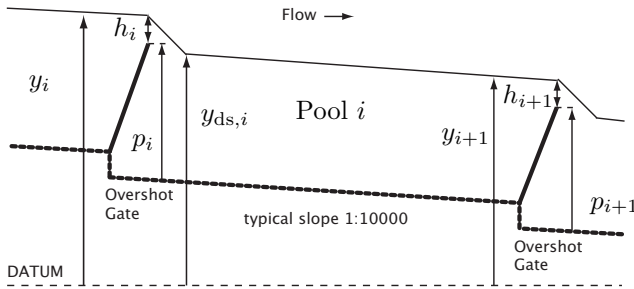


Fig. 3. Stretch of an open-water channel with over-shot gates



Fig. 4. Photo of Gate 5

respectively, p_i and p_{i+1} are the positions of Gates i and $i + 1$, and h_i and h_{i+1} are the heads over gates which are the heights of water above the gates.

There are two gates at each site as shown in Fig. 4. Both gates operate in parallel, i.e. they always have the same position. The water levels and gate positions are the measured variables, and the heads are computed from these measurements. Water levels are measured using submersible level pressure sensors and gate positions are measured based on the length of the steel cable between the gates and the motors that move the gates. As the channels are located in rural areas, electric power is supplied by solar panels and data communication takes place via a radio network, see Fig. 4. At each gate there is a micro-processor which processes local information and information communicated over the radio network. In this

work we will focus on Pool 4 and 5 and we will treat the measured offtakes to farms as unknown leaks. Data such as length, width etc. are given in Table 1.

Table 1. Data for Pool 4 and 5

Pool	4	5
Length,	943m	1275m
Bottom width	5.72m	5.18m
Side slope	2	2
Bottom slope $\times 10^{-4}$	3.4483	2.6667
Upstream gate width	1.91m	1.91m
Downstream gate width	1.91m	1.91m
$c_{i,in}$	0.0341	0.0200
$c_{i+1,out}$	-0.0341	-0.0200
τ_i	4min	6min

2.1 Models for leak detection

For detection purposes a simple volume balance (Weyer (2001)) describes the relevant dynamics well. The volume balance is given by

$$\frac{dV}{dt} = Q_{in}(t) - Q_{out}(t)$$

where V is the volume of the pool and Q_{in} and Q_{out} are the inflow and outflow respectively. The flow over an overshoot gate in free flow can be approximated by (Bos (1978)) $Q(t) = ch^{3/2}(t)$ where h is the head over gate and c a proportionality constant.

Assuming that the volume of water in a pool is proportional to the downstream water level, we obtain the following model for Pool i

$$\dot{y}_{i+1}(t) = c_{i,in}h_i^{3/2}(t-\tau_i) + c_{i+1,out}h_{i+1}^{3/2}(t) - d_i(t) - l_i(t) \quad (1)$$

where $d_i(t)$ represents known offtakes to farms and side channels in Pool i , and $l_i(t)$ represents the effects of leaks and unknown offtakes¹. A time delay τ_i has also been introduced to take into account the time between water passes the upstream Gate i and the effects reaches the downstream Gate $i + 1$ where y_{i+1} is measured. As most offtakes are at the downstream end of a pool, there is no time delay in the $d_i(t)$ term. The $l_i(t)$ term is unknown and may be a (scaled) time delayed version of an actual leak. This is unimportant as far as detection is concerned, but it may have to be taken into account if we also want to estimate when a leak started.

By replacing the derivative $\dot{y}_{i+1}(t)$ by the difference $(y_{i+1}((k+1)T) - y_{i+1}(kT))/T$ where T is the sampling interval, a discrete time model can be obtained. Here a sampling interval of 1 minute was used and hence we obtained the model

$$y_{i+1}(k+1) = y_{i+1}(k) + c_{i,in}h_i^{3/2}(k - \tau_i) + c_{i+1,out}h_{i+1}^{3/2}(k) - d_i(k) - l_i(k) \quad (2)$$

The parameters $c_{i,in}$, $c_{i+1,out}$ and τ_i were estimated from observed data using prediction error methods for system identification (Weyer (2001), Ooi et. al. (2005), Eurén and Weyer (2006)). The parameters are given in Table 1.

¹ For the purpose of this study we have not explicitly considered the effects of rainfall and evaporation. This is often reasonable since rainfall, if measured, can be incorporated in $d_i(t)$, and losses due to evaporation are often small compared to losses due to leaks.

3. CUSUM ALGORITHM FOR LEAK DETECTION

Equation (2) expresses the change in water level as a function of the in- and outflows. Intuitively one can therefore detect leaks by comparing the change in water level against the known in- and outflows. In the absence of leaks equation (2) can be written as

$$y_{i+1}(k+1) - y_{i+1}(k) - c_{i,\text{in}}h_i^{3/2}(k - \tau_i) - c_{i+1,\text{out}}h_{i+1}^{3/2}(k) + d_i(k) = 0 \quad (3)$$

and when a leak is present the left hand side of (3) becomes negative. With this observation the leak detection problem becomes a standard change detection problem. The variable

$$z_i(k+1) = y_{i+1}(k+1) - y_{i+1}(k) - c_{i,\text{in}}h_i^{3/2}(k - \tau_i) - c_{i+1,\text{out}}h_{i+1}^{3/2}(k) + d_i(k) \quad (4)$$

takes on values around zero before a change, and negative values after a change, i.e. when a leak is present. Note that all variables in (4) are measured or known so $z_i(k)$ can be computed on-line.

One of the most well known algorithms for this problem is the cumulative sum (CUSUM) algorithm (Page (1954), Basseville and Nikiforov (1993), Gustafsson (2000)). If a leak is present

$$g_i(l) = \sum_{k=\tau_i+1}^l z_i(k)$$

will become more and more negative as l increases and an alarm is raised when a threshold is exceeded. In implementations of the CUSUM algorithm it is common to reset $g_i(l)$ to zero if it becomes positive and to add a small positive drift term to $z_i(k)$. As a rule of thumb (Gustafsson 2000) the drift term should be chosen equal to half the change we want to detect. The CUSUM algorithm can therefore be written as

```

gi(τi) = 0
for l = τi, τi + 1, ...
    gi(l + 1) = min(gi(l) + zi(l + 1) + ai, 0)
    if gi(l + 1) < γi
        Alarm time = l + 1
        gi(l + 1) = 0
    end %if
end %for
    
```

An alarm is raised if $g_i(l+1) < \gamma_i$, and a_i in (5) is the drift term. The threshold γ_i is a trade off between quick detection of leaks and false alarm rate. If $-\gamma_i$ is small, leaks will be detected quickly, but the false alarm rate will also be high. The last reset time before the first detection can be taken as a lower estimate for the time when the leak started since there is no indication ($g_i(l) + z_i(l+1) + a_i$ was positive) that a leak was present at that time.

3.1 Robustness with respect to uncertainty in the model parameters

The CUSUM algorithm is quite robust to uncertainty in the model parameters since such uncertainty can be compensated for by adjusting the threshold value as shown next. In our case $c_{i,\text{in}} = -c_{i+1,\text{out}}$ since we have identical

gates at the upstream and downstream end of the channel. Let $c_i = c_{i,\text{in}}$. Ignoring the resets and the drift term, $g_i(l)$ can be written as

$$g_i(l) = y_{i+1}(l) - y_{i+1}(\tau_i) - c_i \sum_{k=\tau_i+1}^l (h_i^{3/2}(k - \tau_i) - h_{i+1}^{3/2}(k)) + \sum_{k=\tau_i+1}^l d_i(k)$$

and the alarm condition becomes

$$\tilde{y}_{i+1}(l) - c_i x_i(l) < \gamma_i \quad (6)$$

where $\tilde{y}_{i+1}(l) = y_{i+1}(l) - y_{i+1}(\tau_i) + \sum_{k=\tau_i+1}^l d_i(k)$ and $x_i(l) = \sum_{k=\tau_i+1}^l (h_i^{3/2}(k - \tau_i) - h_{i+1}^{3/2}(k))$. From (6) it can be seen that an incorrect value of the model parameter c_i can be compensated for by adjusting the threshold γ_i . These days irrigation channels often have automatic control systems for water level regulation and the water level will not vary much and hence $y_{i+1}(l) \approx y_{i+1}(\tau_i)$. If in addition there are no known off takes, the alarm condition becomes approximately

$$-c_i x_i(l) < \gamma_i \quad \text{or} \quad -x_i(l) < \gamma_i / c_i \quad (7)$$

in which case the effect of an incorrect model parameter can be completely absorbed by adjusting the threshold γ_i . (7) is equivalent to monitoring the accumulated flows over the upstream and downstream gate and to raise an alarm if more water flows over the upstream gate than over the downstream gate which makes sense if the water level is constant and there are no known offtakes.

4. EXPERIMENTAL RESULTS

4.1 Operational conditions

The Coleambally Channel no. 6 is operated in closed loop as sketched in Fig. 5. The setpoint for the water level in Pool i is $r_{i+1}(t)$, and the output of the controllers are $u_i(t) = h_i^{3/2}(t)$. The controller equations are given by

$$U_i(s) = C_i(s)(R_{i+1}(s) - Y_{i+1}(s)) + F_i(s)U_{i+1}(s) \quad (8)$$

where $U_i(s)$, $U_{i+1}(s)$, $R_{i+1}(s)$ and $Y_{i+1}(s)$ are the Laplace transform of $u_i(t)$, $u_{i+1}(t)$, $r_{i+1}(t)$ and $y_{i+1}(t)$ respectively. The feedback controllers $C_i(s)$ are PI controllers augmented with lowpass filters, i.e.

$$C_i(s) = \frac{K_i(1 + T_i s)}{T_i s} \cdot \frac{1}{1 + T_{i,f} s}$$

$F_i(s)$ are lowpass filters with gains less than 1 in the pass band. For details see Ooi and Weyer (2007).

4.2 Implementation issues

The measurements required in order to implement the CUSUM algorithm for Pool i are y_{i+1} , h_i and h_{i+1} . This means that the only data communication that has to take place is between Gate $i+1$ and i . As part of the computations for the controller in Fig. 5 y_{i+1} and h_{i+1} are already transmitted to Gate i , and hence the leak detection algorithm can be executed locally at Gate i without any additional data communication. In other words the leak detection algorithm allows for a decentralised implementation requiring only neighbouring gates to communicate. Communication with a central node is only necessary when an alarm is raised.

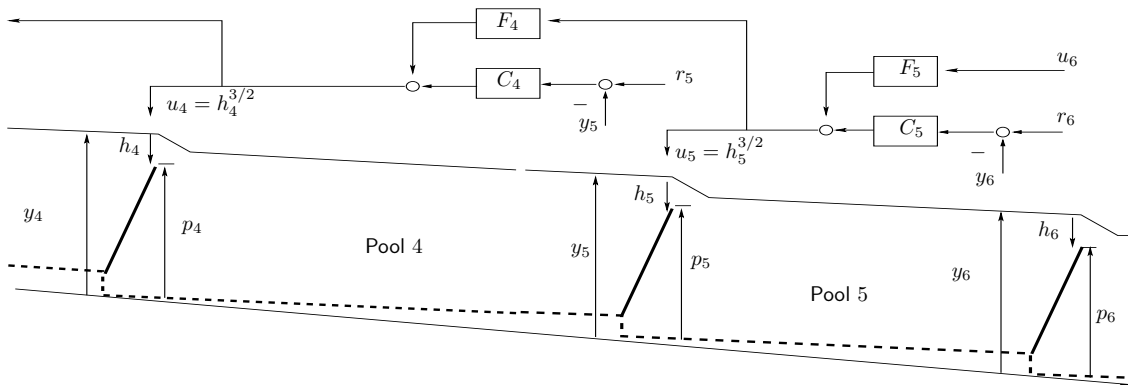


Fig. 5. Controller configuration for Pool 4 and 5

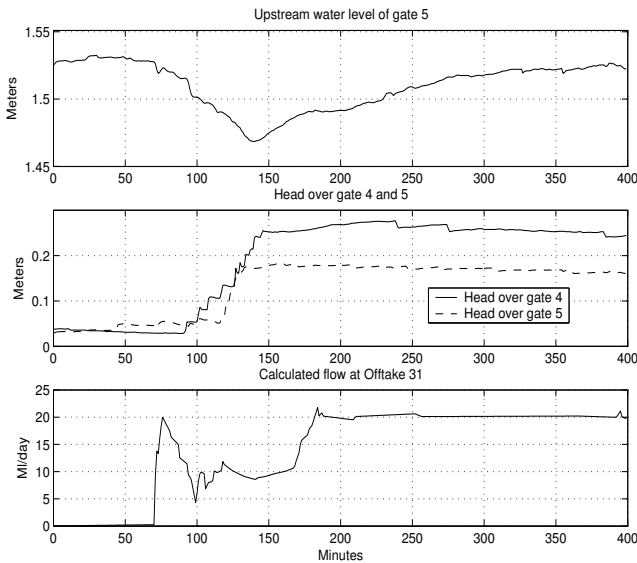


Fig. 6. Data for Pool 4

4.3 Experimental data

The data sets are shown in Fig. 6 and 7. Gate 6 was under manual control and Gate 4 and 5 were controlled as shown in Fig. 5. The flow (head) over Gate 6 was increased at time 110min causing the water levels to drop before the controllers brought them back. In addition as shown in the bottom graph of Fig. 6 an offtake started at time 70min at location 31 which is just upstream of Gate 5. At location 32 which is just upstream of Gate 6 there was a small ongoing offtake which finished at time 280min as shown in the bottom graph of Fig. 7. For the purpose of this study, these offtakes were treated as unknown leaks. The sampling period for the water levels and the gate positions was $T = 1$ min. Note that the water levels and gate positions at a site is relative to a local reference points, and hence the measured water levels at Gate 5 and 6 are not directly comparable. Also note that the head over Gate 5 is the same in Fig. 6 and 7. They look slightly different only because one is drawn with a solid line and the other with a dashed line.

4.4 Results

We applied the CUSUM algorithm in Section 3 to the data in Fig. 6 and 7. The alarm thresholds were set to

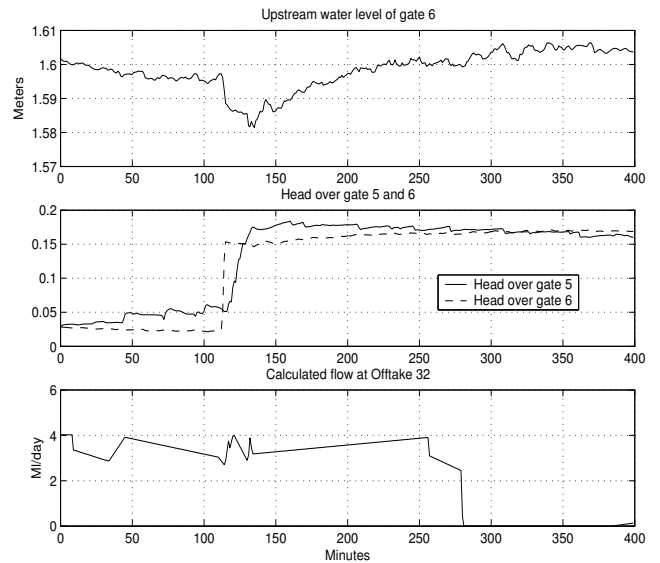


Fig. 7. Data for Pool 5

$\gamma_i = -0.015\text{m}$ for both pools. A rule of thumb is that the drift term in (5) should be selected equal to half the change we want to detect (Gustafsson (2000)). In irrigation channels we often want to detect quite small leaks and therefore $a_i = 0$ is often a reasonable choice. However, if we only want to detect leaks larger than say 1 ML/day ($0.0116\text{m}^3/\text{sec}$) rough calculations based on the surface areas of the two pools suggest that the drift terms should be set to $a_4 = 3.15 \cdot 10^{-5}\text{m}$ for Pool 4 and $a_5 = 2.44 \cdot 10^{-5}\text{m}$ for Pool 5. The cumulative sum $g_i(l)$ with and without the drift term are shown in Figs. 8 and 9 for Pool 4 and 5 respectively. After each alarm $g_i(l)$ was reset to zero, and the detection algorithm was started over again. The dashed vertical lines marks the alarm times, and the dash dotted lines is the last reset time before the first alarm. Of course, in practice corrective action will be taken to stop the leak, and $g_i(l)$ will not be repeatedly reset to zero after an alarm. Here we have done it just to illustrate the algorithms capability of detecting leaks.

As we can see the offtake in Pool 4 is quickly detected both with and without the drift term. It starts at time 70 min and it is detected at time 96 min. This is not surprising since the offtake is relatively large and the algorithm continue to give alarm every 10 minutes or so indicating an ongoing leak. The last reset time is a bit

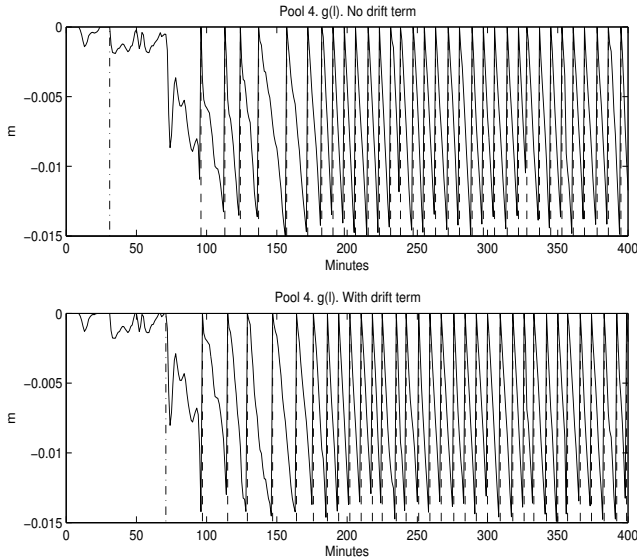


Fig. 8. Results for Pool 4. $a_4 = 0$ in the upper plot. Vertical dashed lines are the alarm times. The vertical dash dotted line (the leftmost vertical line) is the last reset time before the first alarm.

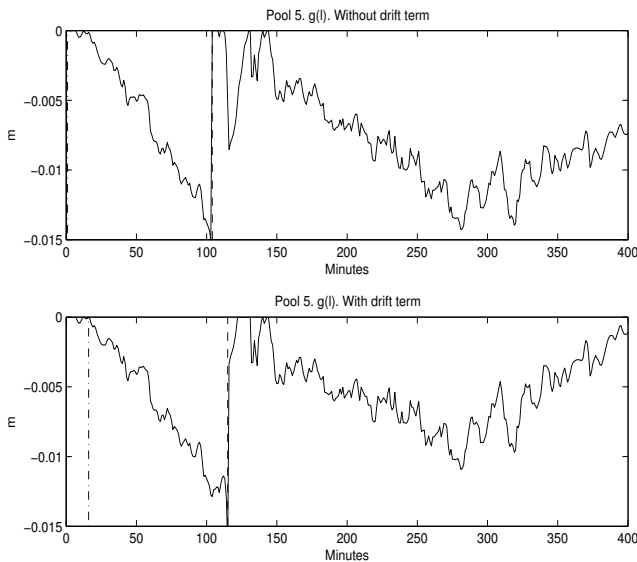


Fig. 9. Results for Pool 5. $a_5 = 0$ in the upper plot. Vertical dashed lines are the alarm times. The vertical dash dotted line (the leftmost vertical line) is the last reset time before the first alarm.

before the actual offtake starts when there is no drift term, but with the drift term the last reset event in Pool 4 before the first alarm occurs at time 70 minutes which is when the offtake starts, and hence provides in this case an accurate estimate of the start time of the leak.

The offtake in Pool 5 is much smaller, and it is detected after about 100 minutes without a drift term and only about 10 minutes later with the drift term. For Pool 5 we only got one alarm which is not surprising since the offtake was small and it also stopped after 280 minutes. Without the drift term the cumulative sum was never reset to 0, but it was reset a few times in the first 15 minutes with the drift term.

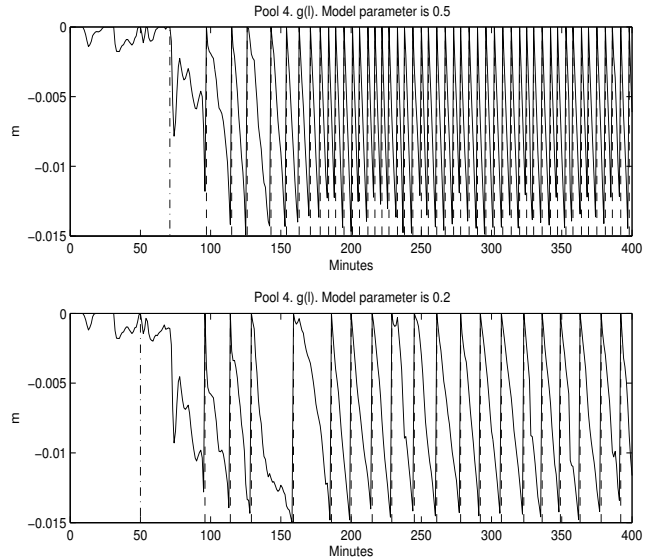


Fig. 10. $g_4(l)$ for Pool 4 with incorrect model parameters. $c_{4,in} = 0.05$ in the upper plot and 0.02 in the lower plot. Vertical dashed lines are the alarm times. The vertical dash dotted line (the leftmost vertical line) is the last reset time before the first alarm.

From Figs. 8 and 9 it can be seen that even quicker detection could have been achieved in both Pool 4 and 5 by using an alarm threshold closer to zero, e.g. $\gamma_i = -0.01m$.

4.5 Robustness against uncertainty in the model parameters.

In order to investigate the robustness we changed the model parameter $c_{4,in} = -c_{5,out}$ for Pool 4 to 0.02 and 0.05. For Pool 5 we changed the value of $c_{5,in} = -c_{6,out}$ to 0.01, 0.012 and 0.028. These values are 40 to 50% above and below the nominal parameter values, so the "errors" are quite large. The drift terms a_4 and a_5 were as before and so were the thresholds γ_4 and γ_5 . The results are shown in Figs. 10 and 11.

Apart from when $c_{5,in} = 0.01$ the leaks are still detected. This shows that the algorithm is robust against uncertainties in the model parameters as shown in Section 3.1. When $c_{i,in}$ is smaller than the nominal value, it takes a longer time for the leaks to be detected. Of course there is a limit to how much uncertainty that can be tolerated, and the leak in Pool 5 was not detected (with the above threshold and drift term) with $c_{5,in} = 0.01$. However, a small adjustment of the threshold to say $\gamma_5 = -0.012m$ (the horizontal line on the bottom graph in Fig. 11) would have been enough to detect the leak. This shows that an error in the model parameter can be partly compensated for by adjusting the threshold.

4.6 Future developments

In addition to detecting the presence of a leak it may also be of interest to estimate the size of the leak and the location within the pool of the leak. A possible approach to the first problem is to estimate the slope of $g_i(l)$ or to compute the average time between alarms. The size of the leak could then be inferred from this information.

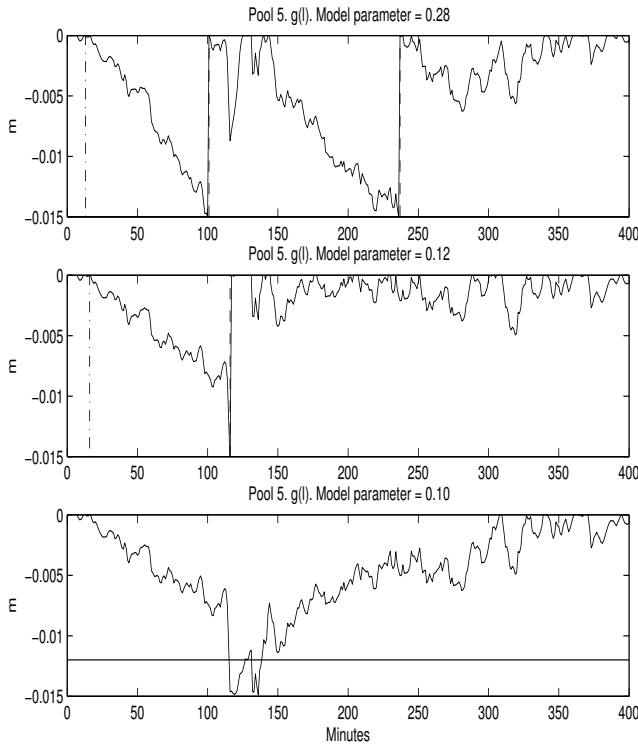


Fig. 11. $g_5(l)$ for Pool 5 with incorrect model parameters $c_{5,in} = 0.028$ in upper plot and 0.012 in the middle and 0.01 in the lower plot. Vertical dashed lines are the alarm times. The vertical dash dotted line (the left most vertical line) is the last reset time before the first alarm.

Based on the time between alarms in the upper plot of Fig. 8 between time 200 minutes and 300 minutes a rough estimate of the size of the leak is 28 ML/day which is a bit larger than the measured value of 20 ML/day. The reason for this can be twofold: Inaccurate measurement of the flow at the offtake and an error in the estimate of $c_{4,in}$ and $c_{5,out}$.

Estimate of the location of the leak is difficult, even with access to measurements of the immediate downstream water level ($y_{ds,i}$ in Fig. 3). Intuitively, if the leak occurs at the downstream end, the water level at the downstream end will start to decrease before the water level at the upstream end, and vice versa if the leak is at the upstream end. However, the natural variations in the measured water levels make it difficult to pinpoint when a water level starts to decrease, and estimates of the location become unreliable.

5. CONCLUSIONS

In this paper we have developed a simple CUSUM algorithm for leak detection in open water channels. The algorithm has been successfully tested on data from an operational irrigation channels with very good results. Leaks were quickly detected, and the algorithm was robust against model uncertainties. Moreover, it allows for a decentralised implementation.

REFERENCES

- Basseville, M., and I.V. Nikiforov (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice Hall
- Bedjaoui, N., X. Litrico, D. Koenig and P.O. Malaterre (2006). " H_∞ observer for time-delay systems Application to FDI for irrigation canals" *Proceedings of 45th IEEE CDC*, pp. 532-537, San Diego, US.
- Bos, M.G. (Ed.) (1978). *Discharge measurement structures*. International Institute for Land Reclamation and Improvement/ILRI, Wageningen, The Netherlands.
- Cantoni, M., E. Weyer, Y. Li, S.K. Ooi, I. Mareels and M. Ryan (2007). "Control of Large-Scale Irrigation Networks". *IEEE Proceedings special issue on The Emerging Technology of Networked Control Systems*, Vol. 95, no. 1, pp. 75-91.
- Dulhoste, J. -F., D. Georges, and G. Besancon (2004). "Nonlinear control of open-channel water flow based on collocation control model." *Journal of Hydraulic Engineering*, pp. 254-266.
- Eurén, K., and E. Weyer (2007). "System identification of irrigation channels with undershot and overshoot gates". *Control Engineering Practice*, Vol. 15, no 7, pp. 813-824.
- Gomez, M., J. Rodellar, and J.A. Mantecon (2002). "Predictive control method for decentralised operation of irrigation canals." *Applied Mathematical Modelling* Vol. 26, pp. 1039-1056.
- Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley.
- de Halleux, J., C. Prieur, J. -M. Coron, B. d'Andréa-Novel and G. Bastin (2003). "Boundary feedback control in networks of open channels", *Automatica*, Vol. 39, no. 8, pp. 1365-1376.
- Litrico X., V. Fromion, J.-P. Baume, C. Arranja and M. Rijo (2005). "Experimental validation of a methodology to control irrigation canals based on Saint-Venant equations", *Control Engineering Practice*, Vol. 13, no. 11, pp. 1425-1437.
- Mareels, I., E. Weyer, S. K. Ooi, M. Cantoni, Y. Li and G. Nair (2005). "Systems engineering for irrigation systems: Successes and challenges. *Annual Reviews in Control (IFAC)*, Vol 29. pp. 191-204.
- Ooi, S.K., M. P. M. Krutzen and E. Weyer (2005). "On physical and data driven modelling of irrigation channels." *Control Engineering Practice* Vol. 13, No. 4, pp. 461-471.
- Ooi, S.K., and E. Weyer (2007). "Control design for an irrigation channel from physical data", *Proceedings of 2007 IEEE Multi-conference on Systems and Control*, pp. 270-275, Singapore, October 2007.
- Page, E.S. (1954). "Continuous inspection schemes". *Biometrika*, Vol. 41, pp. 100-115.
- Schuermans, J., A. Hof, S. Dijkstra, O.H. Bosgra and R. Brouwer (1999). "Simple water level controller for irrigation and drainage canals." *Journal of Irrigation and Drainage Engineering*, Vol. 125. no. 4, pp. 189-195.
- Weyer, E. (2001). "System identification of an open water channel" *Control Engineering Practice* Vol. 9, pp. 1289-1299.
- Weyer E. (2008). "Control of irrigation channels" . To appear in *IEEE Trans on Control Systems Technology*.