DETECTION OF SLUGGISH CONTROL LOOPS IN IRRIGATION CHANNELS [1]

Su Ki Ooi * **E. Weyer** *

* *Cooperative Research Centre for Sensor Signal and Information Processing, Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010 Australia*
Email: {skoo, e.weyer}@ee.mu.oz.au

Abstract: In this paper we apply the algorithm for detection of sluggish control loops developed in (Hägglund, 1999) to irrigation channels. The controller is a PI controller augmented with a low pass filter together with a decoupling term to reduce the interaction between reaches. In simulations, the algorithm is able to distinguish between well tuned controllers and controllers that give sluggish responses. When applied to real data, filtering and deadzones are included in the algorithm to make it more robust. The modified algorithm is then applied to real data from three consecutive reaches of an irrigation channel, and it detects the control loops which need retuning. *Copyright© 2005 IFAC*

Keywords: Performance monitoring of control loops, Control systems, Environmental systems, Irrigation channels

1. INTRODUCTION

Water is a scarce resource, and it is therefore important to manage the water resources well and minimise the losses. This applies particularly to networks of irrigation channels, where huge amounts of water are wasted due to poor management and control. These losses can be reduced by improving the control of the water levels in the channels, see e.g. (Malaterre and Baume, 1998) and the references therein, (Malaterre, 1998), (Schuurmanns *et al.*, 1999), (Litrico, 2002), (Weyer, 2002), (de Halleux *et al.*, 2003), (Weyer, 2003), (Litrico *et al.*, 2003) and (Li *et al.*, 2004).

For decades, the performance of control loops has been monitored by process operators. For a network of irrigation channels, the operators may have to constantly monitor every controlled water level in order to detect deterioration of closed loop performance. To assist the operators, alarms are usually raised when the water level is too high or too low. However, there are many control loops in a network of irrigation channels, and it is very time consuming and even difficult to monitor each and every control loop manually. In addition, automatic design routines such as those in (Ooi and Weyer, 2003) are developed with the purpose of easing and speeding up the process of designing large number of controllers. It would be in conflict with this purpose if one needs to check the performance manually.

It is therefore desirable to have performance monitoring tools that evaluate the performance of the control loops and inform the operator of any badly performing loops. Because experimental access is limited, the performance monitoring tool should be able to detect deterioration in performance using data available from normal day to day operation such as responses due to offtakes of water. In this paper, we consider detection of sluggish control loops, which is one of the most

[1] This work has been supported by the Cooperative Research Centre for Sensor Signal and Information Processing

common effects of badly tuned controllers in irrigation channels.

The paper is organised as follows. In Section 2, a description of the irrigation channel is given. In the following section, the models and the designed controllers are given. A review of the sluggishness detection algorithm is given in Section 4. In Section 5 the algorithm is applied to irrigation channels using simulated data. In Section 6, the algorithm is modified to detect sluggish control loops in three consecutive pools using operational data from an irrigation channel. Some concluding remarks are given in Section 7.

2. CHANNEL DESCRIPTION

The channel considered is the Haughton Main Channel in Queensland, Australia which is automated with overshot gates as shown in Figure 1. We refer to the stretch of the channel between two gates as a pool. We name the pool according to the number of the upstream gate, e.g. Figure 1 shows Pools 8, 9 and 10 which are 1600m, 900m and 3200m long respectively. y_8, y_9, y_{10} and y_{11} are the upstream water level of gates 8, 9, 10 and 11 respectively, and p_8, p_9, p_{10} and p_{11} are the position of gates 8, 9, 10 and 11. The amount of water above the gate is called the head over the gate and denoted by h_8, h_9, h_{10} and h_{11}. The water levels, in mAHD (meter Australia Height Datum), and the gate positions are the measured variables. The head over gate is computed from these variables.

3. CONTROLLERS

The controller considered in (Ooi and Weyer, 2003) is a PI controller augmented with a low pass filter. We refer to this as a PIL controller. The main objective is to reject load disturbances which are offtakes of water from the pools, and integral action is needed in order to achieve this. There are waves present in the channel and the low pass filter is used to suppress these waves. The transfer function of the PIL controller for pool $i-1$ is

$$C_{i-1}(s) = \frac{K_c(1+T_l s)}{T_l s} \cdot \frac{1}{(1+T_f s)} \qquad (1)$$

Properly tuned such controllers give very good performance as demonstrated by the field tests presented in (Weyer, 2002).

A first order nonlinear model, which is derived from a simple mass balance, see (Weyer, 2001), is considered. For Pool $i-1$, we have

$$\dot{y}_i(t) = c_{i-1} h_{i-1}^{3/2}(t-\tau) - c_i h_i^{3/2}(t) + d(t) \qquad (2)$$

where the first term on the right hand side of equation (2) is associated with the inflow to the pool, and the second term with the outflow. $d(t)$ represents an offtake of water.

A new input signal $u_{i-1}(t)$ is introduced and the model (2) becomes

$$\dot{y}_i(t) = c_{i-1} u_{i-1}(t-\tau) \qquad (3)$$

$$u_{i-1}(t) = h_{i-1}^{3/2}(t) - \frac{c_i}{c_{i-1}} h_i^{3/2}(t+\tau) \qquad (4)$$

and the controllers were designed based on this model (3). $u_{i-1}(t)$ depends on future signals, so in practise we use $u_{i-1}(t) = h_{i-1}^{3/2}(t) - \frac{c_i}{c_{i-1}} h_i^{3/2}(t)$. Hence, the total controller with feedforward becomes

$$u_{i-1}(s) = C_{i-1}(s)\left(y_{i,setpoint}(s) - y_i(s)\right)$$

$$h_{i-1}^{3/2}(s) = u_{i-1}(s) + K_{ff,i-1} F_{i-1}(s) \frac{c_i}{c_{i-1}} h_i^{3/2}(s)$$

where we have introduced an additional feedforward gain $K_{ff,i-1}$=0.75 and a second order Butterworth filter $F_{i-1}(s)$ with cut off frequency around half the frequency of the wave in the pool. $h_{i-1}^{3/2}(s)$ and $h_i^{3/2}(s)$ are the Laplace transform of $h_{i-1}^{3/2}(t)$ and $h_i^{3/2}(t)$. Figure 1 shows the side view of the irrigation channel with the controllers where $K_{ff,i-1} F_{i-1}(s) \frac{c_i}{c_{i-1}}$ is denoted as $FF_{i-1}(s)$.

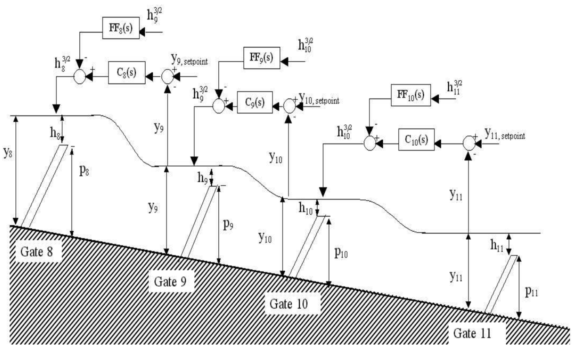

Fig. 1. Side view of irrigation channel with distant downstream controllers with feedforward

From (Ooi and Weyer, 2003), the following discrete time controllers for Pools 9 and 10 were obtained using an automatic tuning routine:

$$C_9(z) = \frac{0.2138(z-0.9710)}{(z-1)(z-0.8535)}$$

$$C_{10}(z) = \frac{0.0661(z-0.9919)}{(z-1)(z-0.9592)} \qquad (5)$$

4. DETECTION OF SLUGGISH CONTROL LOOPS

The main objective of controllers for irrigation channels is to reject load disturbances due to offtakes of water. A well tuned controller should recover from the load disturbances within a short time and bring the water level smoothly back to the setpoint.

(Hägglund, 1999) developed a procedure for automatic detection of sluggish control loops caused by conservatively tuned controllers subjected to load disturbances, and it is briefly discussed next.

4.1 Review of the sluggishness detection algorithm

Figure 2 shows examples of responses of a good and a bad controller subjected to a load disturbance. The good one gives a fast recovery while the badly tuned one gives a sluggish response.

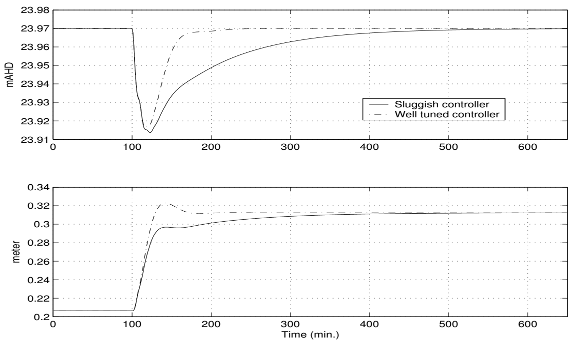

Fig. 2. Good and bad control: process output (top) and control signal (bottom)

The idea behind the algorithm for detection of sluggish control loops is that for sluggish loops, both the process output y and the control signal u will increase slowly for a long time period, see Figure 2. Let Δy and Δu be the increments of y and u respectively. From Figure 2 we can see that initially Δy and Δu for both the good and the bad controllers have opposite signs, i.e. $\Delta u \Delta y \leq 0$. However, after this initial phase, there is a very long time period where $\Delta u \Delta y > 0$ for the sluggish response and this is what (Hägglund, 1999) uses to detect the sluggish response. Based on the time periods for which $\Delta u \Delta y \leq 0$ and $\Delta u \Delta y > 0$, Hägglund formed the Idle index, which is a measure to assess the sluggishness of closed loop responses.

4.2 Idle index computation

Briefly, the procedure to compute the Idle index is as follows. First, compute the time periods t_{pos} and t_{neg} when $\Delta u \Delta y$ are positive and negative. The following procedure is updated every sampling instant.

$$t_{pos} = \begin{cases} t_{pos} + T_s & \text{if } \Delta u \Delta y > 0, \\ t_{pos} & \text{if } \Delta u \Delta y \leq 0, \end{cases}$$

$$t_{neg} = \begin{cases} t_{neg} + T_s & \text{if } \Delta u \Delta y < 0, \\ t_{neg} & \text{if } \Delta u \Delta y \geq 0, \end{cases}$$

where T_s is the sampling period. The Idle index I_i is

$$I_i = \frac{t_{pos} - t_{neg}}{t_{pos} + t_{neg}} \qquad (6)$$

(Hägglund, 1999) also provides a recursive procedure for computing I_i online. For simplicity, we only consider the offline case. I_i is in the interval $[-1, 1]$. Hägglund suggested that an Idle index close to zero indicates that the controller is tuned reasonably well. A sluggish control loop gives I_i closed to 1, while I_i close to -1 may be obtained for a well tuned control loop. However, Idle indices close to -1 are also obtained in oscillatory control loops, see (Hägglund, 1999) for details. Hence, it is desirable to combine the Idle index calculation with an oscillation detection procedure. However, we limit ourself to the detection of sluggish loops in this paper.

In the following simulation, the above algorithm is applied to irrigation channels which are subjected to load disturbances (offtakes), and the times of the offtakes are assumed known.

5. DETECTION OF SLUGGISH CONTROL LOOPS IN IRRIGATION CHANNELS: SIMULATION

Before we proceed, we first discuss the applicability of the algorithm to irrigation channels. Figure 3 shows $y_{10}(t)$, $u_9(t)$ and $h_9(t)$ for Pool 9 with sluggish and well tuned controllers with feedforward subjected to an offtake of water. The downstream gate position is fixed throughout the simulation, see Subsection 5.1 for details on the simulation. Conventionally, $u_9(t)$ is the 'control signal'. However, from Figure 3, we see that after an initial phase, $u_9(t)$ is decreasing, also for the sluggish controller, for a long time period instead of increasing. This is due to the feedforward action, where an extra compensation is added in order to rapidly account for the water flowing out of the channel at the downstream end. This behavior in $u_9(t)$ is not observed if there is no feedforward in the control loop.

In order to apply the proposed algorithm to irrigation channels, instead of using $u_9(t)$ as the control signal we use $h_9(t)$. Now, let Δy_{10} and Δh_9 be the increments of $y_{10}(t)$ and $h_9(t)$ respectively. From Figure 3, during an initial phase $\Delta y_{10} \Delta h_9$ is negative. For the sluggish controller $\Delta y_{10} \Delta h_9 > 0$ for a longer period of time after the initial phase. This is in agreement with the ideas put forward in (Hägglund, 1999) and causes the Idle index to be closer to $+1$ for the sluggish controller.

5.1 Simulation

In this section the algorithm to detect sluggish control loops is applied to Pools 9 and 10 using simulated data. The sampling period is 1 minute. In order to make a comparison, we have intentionally retuned the controllers to give sluggish responses by moving the zeros of $C_9(z)$ and $C_{10}(z)$ to 0.9850 and 0.9950 respectively. An accurate model obtained in (Weyer, 2001) is used to simulate the true system.

The simulation for Pool 9 is as follows. At time 0min the water level is in steady state at the setpoint 23.97mAHD. At time 200 minute, an offtake takes place. The downstream gate position is fixed throughout the simulation. Figure 3 shows plots of y_{10}, u_9 and h_9 for the controller tuned by the automatic tuning routine (5) and for the one that has been retuned to give a sluggish response.

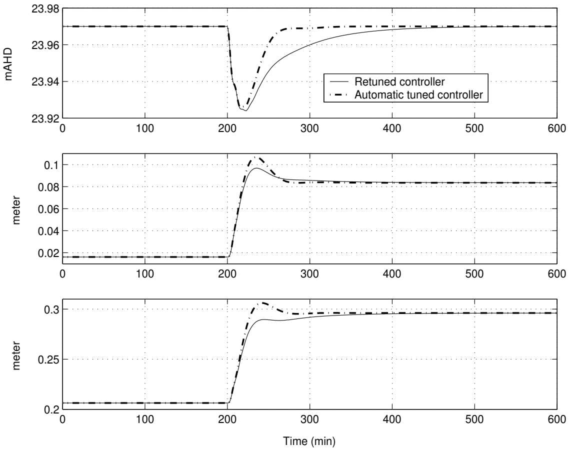

Fig. 3. Pool 9: y_{10} (top), u_9 (middle) and h_9 (bottom)

For Pool 9 we obtained $I_i = -0.1818$ for the controller tuned by the automatic tuning routine and $I_i = 0.8698$ for the sluggish controller. For Pool 10, we have $I_i = -0.1594$ for the automatic tuned controller and $I_i = 0.6881$ for the sluggish one.

5.2 Discussion

The first result to note is that the controllers obtained using the automatic design routine are judged well tuned controllers as $I_i < 0$. From the simulations performed, we can also clearly distinguish between the well tuned controllers and those that give sluggish responses which have I_i closer to 1. As there is a big difference in the Idle index for the well tuned and sluggish controllers in this simulation study, it suggests that one can use zero or a small positive value of the Idle index as threshold value for detection of sluggish loops.

6. DETECTION OF SLUGGISHNESS OF THREE CONSECUTIVE POOLS: REAL DATA

In this section, we consider detection of sluggish control loops in three consecutive pools, Pools 8, 9 and 10 using experimental data. We have two sets of measured data as shown in Figures 4 and 5 (water levels and heads over upstream gate), and they are called the Dec5 and Dec6 data set after the date they were recorded. The sampling period is 1 minute, i.e. $T_s = 1$. The controllers used to obtain these data were designed based on first order linear system identification models and are given by (see (Weyer, 2002) for details)

Table 1. Controller parameters of Pools 8, 9 and 10 used to obtain Dec5 and Dec6 data

Data set	Pool	K_c	T_l	T_f
Dec5	8	2.5	142.86	16.67
Dec5	9	2.0	50.0	10.0
Dec5	10	1.6	142.86	20.0
Dec6	8	3.17	119.05	16.67
Dec6	9	2.0	50.0	10.0
Dec6	10	1.60	113.64	20.0

$$u_{i-1}(s) = C_{i-1}(s)\left(y_{i,setpoint}(s) - y_i(s)\right) \qquad (7)$$
$$h_{i-1}(s) = u_{i-1}(s) + K_{ff,i-1}F_{i-1}(s)\frac{c_i}{c_{i-1}}h_i(s)$$

Note that h and not $h^{3/2}$ are used in (7), and the parameters c_i and c_{i-1} are the parameters in the linear models, and hence different from those of (2). The controllers parameters are given in Table 1. The controllers for Pools 8 and 10 on the 6th of December 2002 were retuned to give faster response than those used on the 5th of December. In the experiment the head over gate 11 was kept at constant values and the effect of an offtake was created by increasing the head over gate 11. From Figures 4 and 5 we can see how the disturbance travels upstream.

From Figures 4 and 5, we see that the measured data are noisy. As pointed out in (Hägglund, 1999), the sluggish detection algorithm is sensitive to noise, since Δy and Δh are used. Hence, we filter the data using a second order butterworth filter with cutoff frequency, $f_c = 0.02$Hz.

In addition, there was deadband of 1.5cm imposed on the gate positions, i.e. the gates did not move if they were asked to move less than 1.5cm. The effect of the deadband can be seen in e.g. Pool 8 in Figure 4 from time 200 to 240min, where the head is nearly constant. In order to take the effect of deadband into account, we have modified the algorithm such that if $\Delta h = h(i+1) - h(i)$ is less than a threshold value called the deadzone, it will be taken as zero, and the next Δh will be computed as $h(i+2)-h(i)$. Otherwise, the next Δh will be computed as $h(i+2) - h(i+1)$. Also, before the maximum deviation of water level from setpoint happen, $\Delta y \Delta h = 0$ will be treated as negative. After the maximum deviation, $\Delta y \Delta h = 0$ will be treated as positive.

Using the unfiltered and the filtered data, Idle indices for Pools 8 to 10 were computed. We call the time when we changed the head over gate 11 the Starttime, and Endtime is the time when the water level is within ± 1.5cm of the setpoint. Starttime of the Dec5 and Dec6 data are 105min and 270min respectively. The results are given in Table 2. In this table, we also included the total time from Starttime to Endtime, SE.

Table 2. Idle indices of Pools 8, 9 and 10. f_c =N/A corresponds to unfiltered data, and Deadzone=0 corresponds to using the original algorithm in Section 4.2.

Data	f_c	Dead-	Pool 8			Pool 9			Pool 10		
Set	(Hz)	zone	I_i	Endtime	SE	I_i	Endtime	SE	I_i	Endtime	SE
Dec5	N/A	0cm	-0.2366	302min	197min	0.0196	157min	52min	0.0076	252min	147min
Dec6	N/A	0cm	-0.1961	383min	113min	-0.4884	315min	45min	0.1471	349min	79min
Dec5	N/A	1.5cm	0.2487	302min	197min	-0.0385	157min	52min	0.3878	252min	147min
Dec6	N/A	1.5cm	0.0089	383min	113min	-0.1111	315min	45min	0.2405	349min	79min
Dec5	0.02	1.5cm	0.1731	313min	208min	-0.4769	170min	65min	0.2168	248min	143min
Dec6	0.02	1.5cm	-0.1220	393min	123min	-0.3103	328min	58min	-0.1111	360min	90min

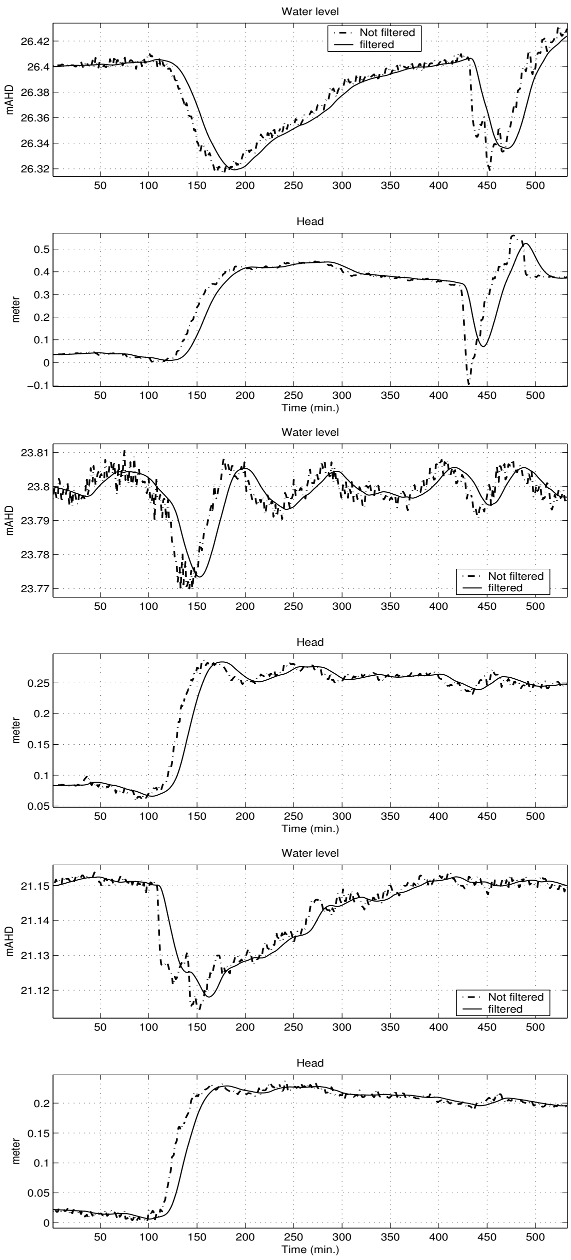

Fig. 4. Dec5 data. Pool 8 (Top), Pool 9 (Middle), and Pool 10 (bottom)

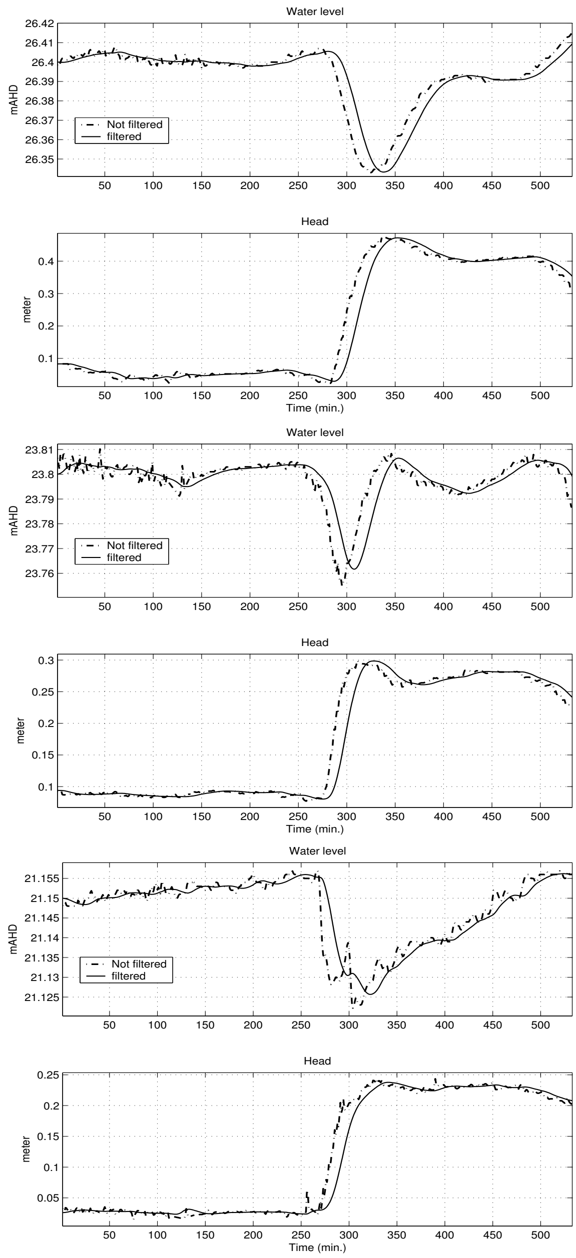

Fig. 5. Dec6 data. Pool 8 (Top), Pool 9 (Middle), and Pool 10 (bottom)

6.1 Discussion

By looking at the data, one would say that the responses on the 5th of December for Pools 8 and 10 are sluggish, while the other responses are quite acceptable.

The effectiveness of the sluggish detection algorithm depends critically on the 'quality'/'smoothness' of the data. From Table 2, with unfiltered data, the Dec6 data set gives larger I_i for Pools 8 and 10 than the Dec5 data set. However, from Figures 4 and 5 it is clear that the controllers used on the 6th of December are faster than those on the 5th. Using the unfiltered data with a deadzone of 1.5cm which has a filtering effect, we have slightly more sensible results in the sense that I_i for Pools 8 and 10 computed using Dec6 data are smaller than those using Dec5 data. However, they are positive.

With filtered data we get results more in line with what we expected. From Table 2, using zero as a threshold the computed I_i can distinguish between sluggish and non-sluggish controllers for Pools 8 and 10. The modified algorithm also gives sensible results for Pool 9 where all I_i's are negative. From Figures 4 and 5, we see that the responses in Pool 9 are good on both days.

Compared with the SE computed in Table 2 for Pools 8 and 10, the I_i's are reasonable in the sense that the longer the SE, the larger the I_i. For Pool 9 with the Dec5 data set, SE=65min and with the Dec6 data set, SE=58min, but the I_i computed for the Dec5 data set is smaller than for the Dec6 data set. However, both I_i computed are negative, hence the results are still sensible.

The proposed procedure assumed that the time of the offtakes are known and hence we know when to compute the Idle index. This is a reasonable assumption in modern irrigation systems where offtakes are measured and/or centrally controlled. In the situation where the times of the load disturbances are unknown, (Hägglund, 1999) suggested that a load detection procedure should be used and I_i is calculated for the period immediately following load disturbances.

7. CONCLUSION

In this paper we have presented some results on detection of sluggish control loops in irrigation channels. In Section 5, the algorithm for detection of sluggish control loops proposed in (Hägglund, 1999) has been applied to irrigation channels using simulated data. From the simulations, we can clearly distinguish between the well tuned controllers and those that give sluggish responses. The sluggish ones have an Idle index close to 1, while the well tuned ones have a negative Idle index.

In Section 6, the algorithm has been modified to detect sluggish control loops in three consecutive pools using measured data. The modified algorithm takes into account the deadband on the gate movement. Also, the raw measured data are noisy and as pointed out in (Hägglund, 1999), the detection algorithm depends heavily on the smoothness of the data. Hence, the measured data are filtered.

From the computed Idle indices, one is able to distinguish between a sluggish and a fast controller. However, one should be careful with blindly applying the results presented here. The data material presented here is limited and more data is required to determine if a cutoff frequency, deadband and threshold can be found such that the algorithm returns sensible results for a broad range of channel types and operational conditions. Nevertheless, the great advantage of this method is its simplicity. No information about the controllers or the systems is required, only the control signal and the process output are needed.

REFERENCES

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**, pp. 1365–1376.

Hägglund, T. (1999). Automatic detection of sluggish control loops. *Control Engineering Practice* **Vol. 7**, pp. 1505–1511.

Li, Y., M. Cantoni and E. Weyer (2004). Design of a centralized controller for an irrigation channel using H_∞ loop-shaping. *Proceedings of the UKACC Control conference, University of Bath, UK*.

Litrico, X. (2002). Robust IMC Flow Control of SIMO Dam-River Open-Channel Systems. *IEEE Trans. on Control Systems Technology* **Vol. 10**(no. 3), pp. 432–437.

Litrico, X., V. Fromion, J.-P. Baume and M. Rijo (2003). Modelling and PI control of an irrigation canal. *Proceedings of the 2003 European Control Conference, Cambridge, UK*.

Malaterre, P. O. (1998). PILOTE: Linear Quadratic Optimal Controller for irrigation Canals. *Journal of Irrigation and Drainage Engineering* **Vol. 124**(no. 4), pp. 187–194.

Malaterre, P.-O. and J.-P Baume (1998). Modeling and regulation of irrigation canals: existing applications and ongoing researches. *IEEE International Conference on Systems, Man and Cybernetics, San Diego, California* pp. 3850–3855.

Ooi, S. K. and E. Weyer (2003). Control design for an irrigation channel from physical data. *Proceedings of the 2003 European Control Conference, Cambridge, UK*.

Schuurmanns, 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 Practise* **Vol. 9**, pp. 1289–1299.

Weyer, E. (2002). Decentralised PI controller of an open water channel. *Proceedings of the 15th IFAC World Congress, Barcelona, Spain*.

Weyer, E. (2003). LQ control of an irrigation channel. *Proceedings of the 42nd IEEE Conference on Decision and Control, Hawaii, USA* pp. 750–755.