PRELIMINARY RESULTS FOR DOWNSTREAM WATER-LEVEL FEEDBACK CONTROL OF BRANCHING CANAL NETWORKS

Brian T. Wahlin[1]
Albert J. Clemmens[2]

ABSTRACT

Over the last 40 years researchers have made various efforts to develop automatic feedback controllers for irrigation canals. However, most of this work has concentrated on feedback controllers for single, in-line canals with no branches. In practice it would be desirable to automate an entire canal network and not just one of the branches. Because the branches in a network are hydraulically coupled with each other, a branching canal network cannot be controlled by designing separate controllers for each branch and then letting them run simultaneously. Changing the gate position in one pool on one branch can affect the water levels in pools on other branches. Because of this effect, the controllers designed for each of the in-line branches of the network will interfere with each other and potentially create instabilities in the branching canal network. Thus, the controller must be designed for the network as a whole and the branching flow dynamics must be explicitly taken into account during the controller design process. This paper presents preliminary simulation results on three different downstream feedback controllers on a branching canal network. The first controller is a series of Proportional-Integral (PI) controllers, one per pool. The second is a fully centralized PI controller. The third controller uses Model Predictive Control (MPC) to determine the appropriate control actions.

INTRODUCTION

The main purpose of an irrigation water delivery system is to deliver water to users at the desired time, rate, frequency, and duration. Most operators of irrigation water delivery systems operate the canals using manual techniques. Routing known flow changes and accounting for unknown flow disturbances and flow measurement errors using manual control is a difficult and time-consuming process. Thus, some canal operators have turned to automatic control techniques in an attempt to more efficiently control irrigation water delivery systems.

Over the past 40 years, researchers have proposed a wide variety of algorithms for automatic control of water levels in irrigation canals (Malaterre *et al.* 1998). These control algorithms include classic proportional-integral (PI) controllers, heuristic controllers, predictive controllers, and optimal controllers. Despite the

[1] Civil Engineer, U.S. Water Conservation Laboratory (USWCL), 4331 E. Broadway Rd., Phoenix, AZ 85040
[2] Laboratory Director, USWCL

large number of proposed algorithms, automatic control of irrigation water delivery systems has been successful under only a limited range of operating conditions (Rogers and Goussard 1998 and Pongput and Merkley 1997).

One drawback to the existing attempts to automatically control irrigation water delivery systems is that these control algorithms are only applicable to single in-line canal systems. If they are applied to branching canal networks, then the controllers will interfere with each other and may adversely impact the overall system. Thus, it is desirable to develop an automatic control system that can be applied to branching canal networks.

This paper presents preliminary simulation results on three different downstream feedback algorithms on a branching canal network. The first algorithm uses Proportional-Integral (PI) control, with each check controlled by errors in the water level at the downstream end of the downstream pool. The second algorithm is the fully centralized PI controller, which is a PI controller with complete hydraulic decoupling that explicitly takes the delay times of the pools into account (see Clemmens and Schuurmans 1999 for more details on this controller). With this controller, the flow rate at any check structure can be influenced by water level errors in any pool and by prior control actions at any check structure. These first two controllers use optimization techniques off-line to determine the controller constants. The third algorithm uses Model Predictive Control (MPC) to determine the appropriate control actions. For this controller, optimization techniques are performed on-line at each control time step. All of these controllers use a lumped-parameter linear approximation of the Saint Venant equations as their underlying linear process model.

BACKGROUND

Linear Process Model

The core of any automatic control system is the underlying process model that is used to model flow in open channels. Open-channel flow is described by the Saint Venant equations, which are a set of hyperbolic, nonlinear, partial differential equations that are distributed in time and space. However, nonlinear feedback control is not as easy to define as linear feedback control. Thus, the control problem is greatly simplified if the process model is linearized and linear feedback control is utilized.

Schuurmans *et al.* (1995) developed the integrator-delay (ID) model to describe flow in open channels. The ID model is a lumped-parameter linear response model that can handle backwater effects as well as normal flow conditions. For one pool, the ID model can be expressed as:

$$\frac{de}{dt}=-\frac{q_{out}(t)}{A_s} \qquad \text{for } t \le \tau \tag{1}$$
$$\frac{de}{dt}=\frac{1}{A_s}\left(q_{in}(t-\tau)-q_{out}(t)\right) \qquad \text{for } t > \tau$$

where e is the deviation of the downstream water level from its desired steady-state level, q_{in} is the deviation of the upstream inflow to the pool from its steady-state value, q_{out} is the deviation of the downstream outflow from the pool from its steady-state value, A_s is the backwater surface area of the pool, τ is the delay time of the pool, and t is time. The ID model depends on only two hydraulic parameters per pool: the delay time, τ, and the backwater surface area, A_s. Once these two properties are determined, the hydraulic characteristics for the pool are completely defined. See Clemmens *et al.* (1997) for details on determining A_s and τ.

State-Space Representation

There are many benefits to using the ID model as the underlying linear process model. First, it is a lumped-parameter model, so it is mathematically easier to handle compared to a distributed model. Second, the ID model can be used to define the discrete state-transition equations commonly used in linear system theory without the need for state estimation techniques:

$$\mathbf{x}(k+1)=\ddot{\mathbf{o}}\mathbf{x}(k)+\tilde{\mathbf{A}}\mathbf{u}(k) \tag{2}$$

$$\mathbf{e}(k)=\mathbf{C}\mathbf{x}(k) \tag{3}$$

where $\mathbf{u}(k)$ is a vector of changes in control actions at time k, $\mathbf{x}(k)$ is a vector of changes in the state of the system at time k, $\mathbf{e}(k)$ is a vector of water level errors at time k, and $\mathbf{\Phi}$, $\mathbf{\Gamma}$, and $\mathbf{C}$ are the state-transition matrices, which are defined by discretizing the ID equations. A convenient way to define the state vector is to have it consist of changes in water level errors, previous incremental control actions, and previous water level errors (Clemmens and Schuurmans 1999).

Controller Objectives

Both the optimal PI controllers and MPC determine the appropriate control actions by minimizing an objective function, J:

$$J = \sum_{k=0}^{\infty} \mathbf{e}(k)^T \mathbf{Q}\mathbf{e}(k) + \mathbf{u}(k)^T \mathbf{R}\mathbf{u}(k) \qquad (4)$$

where **Q** is the penalty function for water level errors and **R** is the penalty function for changes in control actions. Although these controllers are tuned using optimization techniques, the user still needs to determine the appropriate values for the penalty function matrices **Q** and **R**. Typically, this is done using trial-and-error techniques. To simplify this process, **Q** is set equal to **I**, and **R** = ρ **P** where ρ is the penalty weight for the controller and **P** is the identity matrix weighted by the relative capacity of each pool in the canal (see Clemmens *et al.* 1997 for more details).

Optimal Proportional-Integral Controllers

Using linear system theory, the typical control law for PI controllers in state-space form can be expressed as:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) \qquad (5)$$

where **K** is the controller gain matrix determined by minimizing the objective function, *J*. The solution for **K** is subject to the dynamic characteristics of the physical system, as described by the state-transition equations (2) and (3). This optimization procedure is performed once and the same gain matrix, **K**, is used throughout the simulation. The form of **K** defines the type of PI controller that is used. For example, if all of the elements of **K** are nonzero, then this represents the fully centralized PI controller. If only the elements of **K** that correspond to the proportional and integral constants of the controller are nonzero and the rest of the elements are zero, then this form of **K** represents a series of simple PI controllers, one per pool (see Clemmens *et al.* 1997 for more details). For these optimal PI controllers, the value of ρ needs to be determined through trial-and-error techniques.

Model Predictive Control

MPC originated in the late 1970s and has been used extensively in the process control industry. MPC has three basic components, summarized as follows (Camacho and Bordons 1999):

1. A linear process model, the ID model in this case, is used to predict the system output for some time into the future, □(*k*+*i*|*k*). The time that is predicted into the future is called the prediction horizon, *p*. The output predictions have two components: a free response and a forced response (Clarke 1994). The free response is the expected behavior of the system

assuming no future control actions. The forced response is the additional component of the process output that is due to the unknown future changes in control actions, **u**(k+i|k). The forced response is considered over a time horizon called the control horizon, m, while the free response is considered over the entire prediction horizon, p. The control horizon is less than or equal to the prediction horizon. While in the control horizon, the process model is used to obtain output predictions based on both the free and forced responses. After the control horizon has passed, the remaining output predictions are based on only the free response of the system. This prediction strategy is shown in **Figure 1**.

2. An objective function, similar to equation (4), is minimized by adjusting the future control actions, **u**(k+i|k). This optimization problem is subject to the many constraints that may be imposed on the system.
3. Once the sequence of future control actions that minimizes the desired objective function is determined, only the first set of control actions is implemented on the system, **u**(k+1|k). The system is then updated and the process is repeated. This is known as the receding horizon strategy.

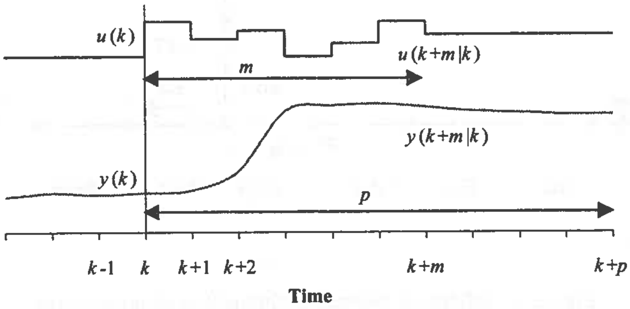

Figure 1. MPC prediction strategy (from Camacho and Bordons 1999)

MPC differs from the optimal PI controllers in that there is no explicit control law such as equation (5), and the optimization problem is solved on-line at each time step during the simulation. Implementing MPC on-line may present some difficulties because the optimization problem may be very complex and require an extensive amount of computing power to solve. Also, feasibility is an issue with MPC. If the constraints imposed on the optimization problem are too restrictive, then the problem may become infeasible and the controller will not function. Tuning the MPC controller consists of determining the appropriate values for the control horizon, the prediction horizon, and the penalty weight for the controller (ρ).

CONTROL OF BRANCHING CANAL NETWORKS

The first step in developing a controller for a branching canal network is to make sure that the state-transition equations capture the branching canal hydraulics by modifying the underlying ID equations. In the pool where the branch occurs, equation (1) must be modified to include the outflow into both branches of the canal system. This methodology is best explained by looking at an example of a branching canal system. Consider ASCE test canal 1 (see Clemmens *et al.* 1998 for more details on the ASCE test canals) and assume that there is a branch that occurs at the downstream end of pool 4. One of the branches contains pools 5 and 6 from the ASCE test canal 1 while the other branch contains pools 7 and 8, as shown in **Figure 2.**

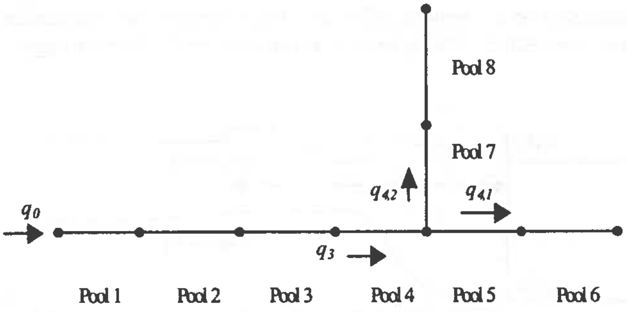

Figure 2. Schematic diagram of branching canal network

Without the branch, the ID equation for pool 4 can be expressed as:

$$\frac{de_4}{dt} = \frac{1}{A_{s4}}\left(q_3(t-\tau_4) - q_4(t)\right) \tag{6}$$

When a branch is present at the end of pool 4, the underlying ID equation is modified to become:

$$\frac{de_4}{dt} = \frac{1}{A_{s4}}\left(q_3(t-\tau_4) - q_{4,1}(t) - q_{4,2}(t)\right) \tag{7}$$

The ID equations for pools 5 and 7 can be developed with q_{in} equal to $q_{4,1}$ and $q_{4,2}$, respectively. The ID equations for the remaining five pools retain the form of equation (6). The eight ID equations are then discretized and placed in the state-space form of equations (2) and (3) (see Clemmens and Schuurmans 1999 for more details on how this is done). Once the proper modifications have been made to the underlying ID equations, optimal control theory or MPC can be applied to the system. For the optimal PI controllers, an additional adjustment needs to be made. Elements in the gain matrix, **K**, which are not feasible for the branching canal network, need to be set to zero. For example, for a fully centralized PI controller, a portion of the control actions at gate 7 would be passed to gate 5. For an in-line system, this is appropriate. However, for the system shown in **Figure 2**, gate 7 is hydraulically isolated from gate 5 and it would not be appropriate to pass a portion of the control actions at gate 7 to gate 5. Thus, these infeasibilities need to be identified in the branching canal network and the corresponding elements in the gain matrix need to be set to zero.

Control Tests

The authors performed simulations on the branched version of test canal 1 using the same initial conditions specified for test case 1 (see Clemmens *et al.* 1998 for details on ASCE test case 1). To test the effectiveness of the controller, an offtake change occurred in each section of the branching canal network (*i.e.*, upstream from the branch and in each of the two branches). Six hours into the simulation, the offtake flows at pools 3 and 8 increased from 0.1 m^3/s to 0.2 m^3/s, while the offtake at pool 5 was shut off. Simulations were carried out using the simple PI controller, the fully centralized PI controller, and MPC. All simulations were performed using the hydrodynamic model SOBEK (Delft Hydraulics 2000). SOBEK has the ability to simulate branching canal networks and to be linked to MATLAB (MathWorks 2000). All of the control routines were written as MATLAB m-files that interfaced with SOBEK. Because the goal of this paper was to determine the feasibility of these types of feedback controllers on branching canal networks, several simplifications were made to the ASCE test case 1. The simulations were performed only under tuned conditions, the minimum gate movement constraints were not enforced, and all of the flow changes were considered unscheduled (*i.e.*, no feedfoward routine was implemented).

Two constraints were imposed on the simulations: 1) the gates were not allowed to be completely closed and 2) the gates were not allowed to be taken out of the water. For the optimal PI controllers, these constraints were imposed after the control calculations were performed. In other words, the control law was used to determine a set of changes in control action variables. If these changes caused the constraints to be violated, then the control actions were adjusted until the constraints were satisfied. For MPC, the constraints were written explicitly into the constrained optimization problem.

RESULTS

Optimal Proportional-Integral Controllers

From past experience, the authors found that setting $\rho = 20$ works well for the steep test canal 1. Because no control actions are passed to other gates for the simple PI controller, no further modifications need to be made to the gain matrix and constrained optimization techniques can be used to determine the coefficients of **K**. About 10 hours after the disturbances, the PI controller returned the water levels to their setpoints and had a maximum deviation from the setpoint of about 0.2 m (**Figure 3**). Overall, these results are similar to other simulation results obtained using simple PI controllers on the unscheduled flow changes for test case 1-1 on the in-line test canal 1. For example, both Clemmens and Wahlin (1999) and Wahlin and Clemmens (1999) present simulation results that have similar maximum water level deviations for the unscheduled portion of the test case 1-1 as reported here for the branching canal network. However, the PI controllers for the in-line test canal 1 were not as sluggish as the PI controller for the branched system, and the water levels were returned to their setpoints in about six hours. In practice, these water level deviations and settling times may not be acceptable. Prescheduled delivery changes with a feedforward routine would improve the overall performance of this controller.

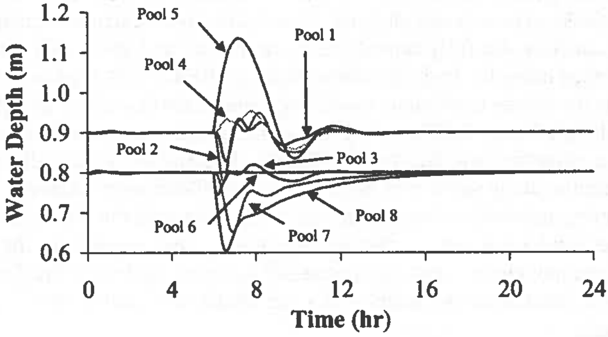

Figure 3. Simulation results using the simple PI controller ($\rho = 20$)

For the fully centralized PI controller, the infeasibilities that occur due to the branching canal network dynamics were identified and the corresponding elements in the gain matrix were set to zero. The gain matrix was then determined using constrained optimization techniques. There is a marked improvement in using the fully centralized PI controller (**Figure 4**) over the simple PI controller (*e.g.*, less overshoot, less oscillations, faster settling time, *etc.*). Within six hours of the disturbances, the fully centralized PI controller restored the water levels to their setpoints. These results agree fairly well with the unscheduled simulation results for the fully centralized PI controller on the in-line test canal 1 (Clemmens and Wahlin 1999). Again, overall controller performance would be improved with the addition of a feedforward routine.

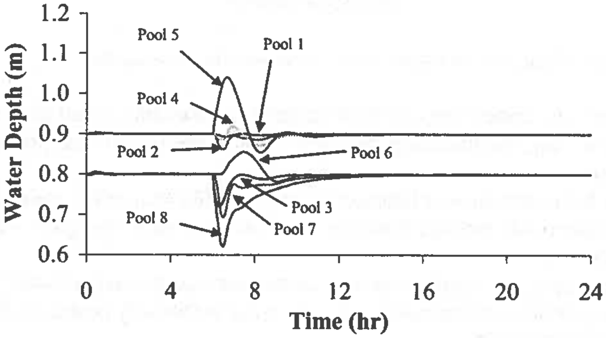

Figure 4. Simulation results using the fully centralized PI controller ($\rho = 20$)

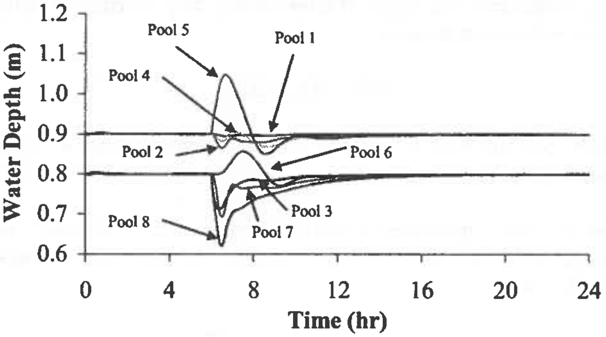

Figure 5. Simulation results using Model Predictive Control ($\rho = 10$)

Model Predictive Control

For the MPC simulations, the control horizon, m, was set to 20 time steps while the prediction horizon, p, was set to 40 time steps. Unlike the optimal PI controllers, better results were obtained with $\rho = 10$ instead of 20. The simulation results for MPC (**Figure 5**) are similar to those for the fully centralized PI controller, and, within six hours of the disturbances, the MPC controller restored the water levels to their desired setpoints. One of the benefits of MPC is that it has a feedforward routine built into the algorithm. Utilizing this option, the overall performance of the controller would improve.

CONCLUSIONS

Several conclusions can be drawn from these simulation studies:

1. Automatic controllers can be developed for branching canal networks by considering the branching dynamics in the underlying linear process model.
2. The fully centralized PI controller and the MPC controller adequately controlled this simple branching canal network under the given flow conditions.
3. The simple PI controller worked on the branching canal network; however, its performance was not as good as the fully centralized PI controller or MPC.
4. The controller performance was almost identical for the fully centralized PI controller and MPC.

The performance of these controllers on this simple branching canal network is encouraging. Additional simulation studies need to be performed to better define the robustness of these controllers.

REFERENCES

Camacho, E.F., and Bordons, C. (1999). *Model predictive control.* Springer-Verlag, London, UK.

Clarke, D.W. (1994). "Advances in model-based predictive control." *Advances in model-based predictive control*, D.W. Clarke, ed., Oxford University Press, Oxford, UK, 3-21.

Clemmens, A.J., Bautista, E., and Strand, R.J. (1997). *Canal automation pilot project: Phase I report.* WCL Report #22, U.S. Water Conservation Laboratory, Phoenix, AZ.

Clemmens, A.J., Kacerek, T.F., Grawitz, B., and Schuurmans, W. (1998). "Test cases for canal control algorithms." *J. Irrig. and Drain. Engr.*, ASCE, 124(1), 23-30.

Clemmens, A.J., and Schuurmans, J. (1999). "A class of optimal feedback canal controllers, including proportional-integral as a limiting case." *Proc., Workshop on Modernization of Irrigation Water Delivery Systems*, U.S. Committee on Irrigation and Drainage, Phoenix, AZ, 423-438.

Clemmens, A.J., and Wahlin, B.T. (1999). "Performance of various proportional-integral feedback canal controllers for ASCE test cases." *Proc., Workshop on Modernization of Irrigation Water Delivery Systems*, U.S. Committee on Irrigation and Drainage, Phoenix, AZ, 501-516.

Delft Hydraulics. (2000). *SOBEK-flow-module technical reference guide, ver. 2.2.* Delft, the Netherlands.

Malaterre, P.O., Rogers, D.C., and Schuurmans, J. (1998). "Classification of canal control algorithms." *J. Irrig. and Drain. Engrg.*, ASCE, 124(1), 3-10.

MathWorks, Inc. (2000). *MATLAB user's guide, ver. 6.* Natick, MA.

Pongput, K., and Merkely, G.P. (1997). "Comparison and calibration of canal gate automation algorithms." *J. Irrig. and Drain. Engrg.*, ASCE, 123(3), 222-225.

Rogers, D.C., and Goussard, J. (1998). "Canal control algorithms currently in use." *J. Irrig. and Drain. Engrg.*, ASCE, 124(1), 11-15.

Schuurmans, J., Bosgra, O.H., and Brouwer, R. (1995). "Open-channel flow model approximation for controller design." *Appl. Math. Modeling*, 19, 525-530.

Wahlin, B.T., and Clemmens, A.J. (1999). "Performance of several historic canal control algorithms on the ASCE test cases." *Proc., Workshop on Modernization of Irrigation Water Delivery Systems*, U.S. Committee on Irrigation and Drainage, Phoenix, AZ, 467-481.