CANAL AUTOMATION SYSTEM DEMONSTRATION AT MSIDD

A.J.Clemmens[1]
R.J. Strand[2]
L. Feuer[3]
B.T. Wahlin[4]

ABSTRACT

A new canal automation system, known as SACMAN (Software for Automated Canal MANagement), has been developed at the U.S. Water Conservation Laboratory in cooperation with Automata, Inc. through a Cooperative Research and Development Agreement. SACMAN works with a commercial Supervisory Control And Data Acquisition (SCADA) system. It allows canal operators to automatically route scheduled changes in demand through their canal system utilizing volume-compensation and time delay calculations. SACMAN can automatically maintain constant water levels on the upstream side of check structures with either downstream or upstream control logic. SACMAN is also capable of automatically maintaining constant gate flows and making incremental gate flow changes. The operator can also make manual changes to the system without turning the automation off. The SACMAN system is currently being tested in real time on the WM canal, a lateral canal of the Maricopa Stanfield Irrigation and Drainage District (MSIDD) in Central Arizona. At the control center, the SACMAN system uses a standard personal computer and commercial SCADA package. Each gate is operated with limitorque motors (not part of this package), which are controlled with Automata's "Mini" Remote Terminal Unit (RTU). Control is based on water level and gate position sensors. A new gate-position sensor was developed that includes both absolute (potentiometer) and very fine relative (optical encoder) position. Communication between the personal computer and RTUs is accomplished with spread-spectrum radios and MODBUS communication protocol. The entire system is available through Automata, Inc. The paper includes a brief description of the software, hardware, and field-test results.

---

[1] Laboratory Director, U.S. Water Conservation Laboratory, USDA-ARS, 4331 E. Broadway Rd., Phoenix, AZ 85040 bclemmens@uswcl.ars.ag.gov

[2] Electrical Engineer, U.S. Water Conservation Laboratory, USDA-ARS, 4331 E. Broadway Rd., Phoenix, AZ 85040 bstrand@uswcl.ars.ag.gov

[3] President, Automata, Inc., 104 New Mohawk Rd. Suite A, Nevada City, CA 95959-3261 automata@automata-inc.com

[4] Civil Engineer, U.S. Water Conservation Laboratory, USDA-ARS, 4331 E. Broadway Rd., Phoenix, AZ 85040 bwahlin@uswcl.ars.ag.gov

INTRODUCTION

The objective of the canal operator is to provide the correct amount of water to each user, which requires the correct divisions in flow at bifurcations. A manual, local canal operator uses water level deviations to judge whether conditions have changed and whether adjustments are needed. A local canal operator cannot see the entire canal at once and does not know what changes will be felt at that location in the future. A supervisory canal operator can see the entire canal at once and thus can see how flow mismatches vary throughout the system, as reflected by deviations in water levels. Most supervisory canal operators make changes based on these mismatches and then wait for conditions to stabilize, often three or four hours. Any new changes in demand (or inflow outflow difference) during that time are hard to detect since the impact of prior control changes have not yet been felt. Automatic supervisory control overcomes this limitation by making more-or-less continuous adjustments (for large canals, maybe every half hour or so).

In most canal systems, only large main canals are controlled by supervisory control systems, with the rest of the distribution system handled locally, manually. Local automatic upstream water level controllers have shown good performance in controlling water levels, but at the expense of fluctuating downstream flows. Simulation studies of centralized water level control suggest that automatic control can improve water level control on the Arizona Canal compared to manual supervisory control (Clemmens et al 1997, 2001). This has not been demonstrated on the real canal. Even so, improved control of main canals does not solve all water distribution problems. Water users are demanding more flexibility in the timing of water delivery, more constant flows, and more flexibility in adjusting flow rates during irrigation events. Conversion from surface irrigation to pressurized irrigation also requires better control of lateral canals. Districts are facing the need to add field staff to accommodate these user demands.

In the past, SCADA systems have been relatively expensive to install and operate. Costs have come down by nearly an order of magnitude over the last decade, making the automation of lateral canals more feasible. There are many good SCADA packages available for personal computers. Also, remote terminal units (RTUs) can be purchased for 10% of what was available a decade ago. Spread-spectrum radios have opened up communications without the need for FCC licensing. Transducers have seen similar price reductions. Thus the cost and complexity of automation has come down at the same time water delivery demands have increased.

In this paper, we introduce a canal automation system, SACMAN (Software for Automated Canal MANagement). This system interfaces directly to a commercial SCADA package (currently iFIX Dynamics by Intellution, Inc., but other packages can also be used). Communication and control to RTUs is done by the SCADA package. The SACMAN software runs in parallel with the SCADA system and interfaces with the SCADA database. SACMAN hardware includes low-cost RTUs with spread-spectrum radios, pressure transducers, gate position sensors, and gate control relays, all available through Automata, Inc. This paper describes the system and the results of field trials at the Maricopa Stanfield Irrigation and Drainage District (MSIDD).

CONTROL OBJECTIVES

The SACMAN control system has been developed in a flexible manner so that a variety of control objectives can be attained. It is recognized that sloping canal systems cannot automatically respond to large demand changes regardless of the control logic (i.e., open canals cannot perform like closed pipelines). Major flow changes need to be routed through the canal. With SACMAN, this can be done manually by the operator or automatically by SACMAN itself. For feedback control of water level errors, SACMAN examines water level errors within each pool and the history of prior control actions. From this SACMAN determines the control actions needed to correct the errors according to user-defined objectives. These objectives can vary from centralized control (all gates adjusted based on all water level errors) to local downstream control or local upstream control, with many options in between. More details on the control approach can be found in Clemmens et al. (1997) and Clemmens and Schuurmans (2002).

CONTROL SYSTEM IMPLEMENTATION

Hardware

The hardware for this system consists of water level and gate position sensors, RTUs, gate motor drivers, gate motors, spread-spectrum radios, and a personal computer, as shown in Figure 1.

The Automata "Mini" is used as the RTU, which has a 10-bit processor for analog to digital conversion. For this application, it is set up for 4 digital inputs and 4 digital outputs. Any commercially available water level sensor can be used: however, its range (e.g., 4-20 ma) must be compatible with the digital input of the "Mini" (as ordered).

We developed a new gate position sensor that includes two sensors, one for absolute position and one for relative position. When originally developed, we had only an 8-bit processor and could not get adequate resolution with an absolute position sensor. A rigid gear rack, attached to the gate along its centerline, passes through the gate position sensor enclosure. The gear rack rotates a gear that drives two position sensors: a potentiometer that gives absolute gate position to within 0.004 ft or 1.2 mm (based on a 4 ft span divided into 2^{10} or 1024 parts) and an optical encoder that gives relative gate position to within 0.003 ft or 0.9 mm regardless of span (based on diameter of gear). This interval can be cut in half with additional programming, but this does not seem needed at this time. In principle, any gate position sensor can be used.

Automata has standard circuits for controlling gate motors. The circuit boards generally need to be set up to fit the particular gate motor housing being used, or packaged separately.

Communication between the RTUs and the computer is through 900 MHz spread-spectrum radios with MODBUS protocol.

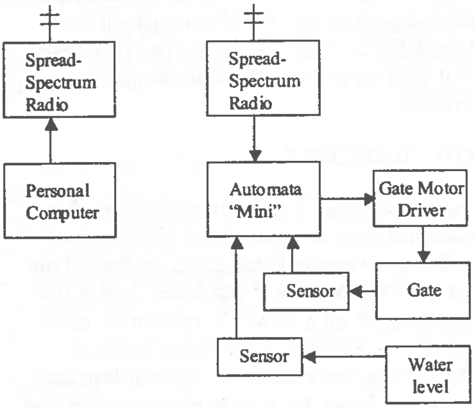

Figure 1. Hardware for SACMAN canal automation control system.

Software

iFIX Dynamics by Intellution, Inc. is the SCADA package currently being used. The canal is set up for supervisory control in a standard manner. iFIX is set up to monitor canal water levels every minute and to store these values in a database. Standard iFIX displays are used to graph the current water levels, flow rates, and gate positions for each check structure. In addition, the water level and flow setpoints are added to the display. The canal operator can still manipulate gates manually.

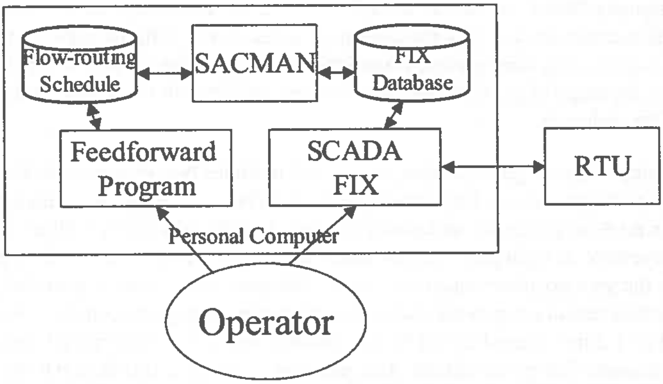

Figure 2. Layout of SACMAN canal automation control system software.

The operator interfaces with the iFIX Dynamics SCADA system to monitor the canal through the iFIX output screens. SACMAN also monitors the canal by reading the iFIX database, as shown in Figure 2. Based on the information in this database, it determines whether control actions are needed. The operator also interfaces with the program for routing demand changes through the canal (feedforward routing). A schedule of offtake flow changes is input by the operator. The program then determines the timing and magnitude of flow changes at all check structures that are required to implement the operator's schedule. This schedule is influenced by the current conditions in the canal, thus information about those conditions must be entered by the users. In the future, this information will be transferred to the feedforward program from the FIX database by SACMAN. The schedule of flow changes is written to a database for SACMAN to read.

SACMAN includes logic: 1) to route intended flow changes through the canal, 2) to adjust gate flow rates in response to downstream water level errors, and 3) to determine gate positions required to provide the desire (setpoint) flow rates at each gate. New values for the flow rate setpoint are determined based on both feedforward routing of intended flow changes and feedback control of downstream water levels. SACMAN currently computes the necessary changes in gate positions for each gate based on the current flow rate, the new flow rate setpoint, and the current gate position. This could also be accomplished at the RTUs by sending just the requested change in flow rate, but this has not yet been implemented. Gate flow rate control uses incremental control, where a change in gate position is determined for the desired change in gate flow. This avoids problems associated with not knowing the flow rates and gate positions exactly. Flow control is accomplished in SACMAN by inverting the gate equations. SACMAN instructs FIX to send information to the RTUs by changing values in the FIX process database.

Feedback control of canal water levels is accomplished with the control logic developed by Clemmens and Schuurmans (2002). It uses a state-space form to compute new check-structure flow rates based on water level errors and prior flow rate changes. Within SACMAN, this amounts to a matrix calculation, with the current states multiplied by a gain matrix to arrive at new flow set points. The coefficients in this matrix are determined with control engineering logic, as described by Clemmens and Schuurmans (2002).

Feedforward routing of intended flow changes is accomplished with software developed by Bautista et al. (2002) based on volume compensation. This requires information for each pool on the relationship between flow rate, water level set point and canal pool volume.

SACMAN uses a constant time step for downstream feedback control of water levels. If it performs the flow control function, the time step for that must be a multiple of the feedback control time step. The raw feedforward schedule is not linked to these time intervals, but should be adjusted by the operator to match either time interval. Eventually, this will be done automatically.

Firmware

The "Mini" uses a 10-bit PIK microprocessor. Codes sent from FIX are used to request sensor information, to change register values, and to cause functions to be performed. The "Mini" is programmed to accept signals in MODBUS protocol. In the current application, a request for a change in gate position is sent as a binary signal. The first bit is a sign bit, which indicates up or down movement. The other seven bits represent the amount of gate movement. This value is placed in a register. Then the relays are set to start moving the gate in the proper direction. For each count on the pulse counter, the register is decremented by one. When it reads zero, the gate motor is stopped. Run-on has never been more than one pulse. A timer limits overrun in the event of sensor failure. The absolute position sensor provides a check, and a backup if the optical encoder fails. Use of this sensor for gate control has not been programmed yet.

APPLICATION AT MSIDD

The SACMAN control system has been implemented on the WM canal at the Maricopa Stanfield Irrigation and Drainage District (MSIDD). The WM canal is a lateral canal with a capacity of 90 cfs (2.5 m^3/s). It was originally supplied with motorized gates. Relay boards, built by Automata, were installed in each gate motor. "Level-tel" water level sensors were installed in existing stilling wells along the upstream side of the gate frame. Automata's new gate position sensors were also installed.

The control logic used in this application is described by Clemmens and Schuurmans (2002). Application to ASCE test canal 1, which is based on the WM canal, is described in Clemmens and Wahlin (2002). The control logic converts water level errors into flow rate changes at each gate. SACMAN determines the gate position change needed to achieve that flow control change and sends a gate position change to FIX. We plan to be able to send flow control commands directly to the RTU, where the change in gate position would be determined.

The current control system determines new flow setpoints for each check structure every 10 minutes. Gate position changes to achieve that flow rate at each check structure are performed every 2 minutes. If a large number of sites are being controlled, the flow control function may best be accomplished locally, depending on the complexity of the flow calculations.

Field Testing

Initial testing of the system was performed in the fall of 1999. Since then, we have converted from Automata's older RTU to the "Mini," the MODBUS protocol was programmed into the "Mini" and Automata's base station firmware, we switched from FM to spread-spectrum radios, and the SACMAN software was totally reworked. These conversions were completed in the summer of 2001. Field studies were limited by infrequent water deliveries along the WM canal. Two successful tests are reported below.

On September 25, 2001, the control system was turned on after a flow increase had been manually routed through the canal via the SCADA system. Only the first 4 pools were under automatic control. Downstream water level control was implemented as a series of simple proportional-integral (PI) controllers (see Clemmens and Schuurmans 2002 for details). A tuning coefficient of $R_1 = 5$ was used to design the controller. The feedback control time step was 10 minutes and the flow control at each check structure was done every 2 minutes. The system was not at steady state when the control system was started and this caused the water levels to oscillate, as shown in Figure 4. The first two pools show considerable oscillations, while the downstream two pools remain nearly constant. Clemmens and Wahlin (2002) determined that $R_1=5$ is too low for this canal, and suggest $R_1 =20$ to avoid these oscillations. At 16:45, flow to turnout M4 was shut off (6.5 cfs or 184 l/s). This shut off was routed through the canal manually, with data generated from the scheduling software (Bautista et al. 2002). The increase in water depth at this time in pools 3 and 4 resulted from improper timing of the flow changes relative to the offtake shutoff.

On October 16, 2001, a similar test was run. In this case, a PI^{+1}_{-1} controller was implemented, tuned with $R_1 = 5$ (Clemmens and Schuurmans 2002). In addition to the standard PI controller, this controller sends control signals to one additional gate upstream and one additional gate downstream. Simulation studies by Clemmens and Wahlin (2002) for the ASCE test canal, which is a simplification of this canal, suggest that this controller was a good compromise between complexity and performance. (For this test, the controller was designed for a ten-minute feedback control interval, but the test was inadvertently run with a two-minute feedback control interval.) Between 11:00 and 12:00, the pressure transducers were recalibrated, creating a disturbance in the controller (Figure 4). This started oscillations in pool 1 which did not dampen quickly. The improvement in controller performance over the simple PI controller in Figure 3 is obvious.

In the spring of 2002, additional testing was done with this control system on the entire canal (8 pools) and for as long as 72 hours continuously. The results of these tests is still being analyzed.

Simulation comparison

The SOBEK unsteady flow simulation model was used to simulate these field tests (Delft Hydraulics 2000). SOBEK was chosen because this canal has many reaches with supercritical flow and few simulation models can handle both supercritical flow and user-defined canal control algorithms. Based on the conditions assumed for controller tuning, the simulation did not match the field measured conditions for the test run on September 25 (not shown). In the simulation, the water levels in pools 1 and 2 did not oscillate as much as the real canal and the jumps in the water levels in pools 3 and 4, when the off take was shut off, do not show up. Clearly, the conditions in the canal are not exactly as assumed. We changed the Manning n values in the simulation from 0.014 to 0.018 to provide untuned conditions. The results are shown in Figure 3. First, the water-level oscillations in pools 1 and 2 now more closely match the observed water level oscillations. Second, the rise in water levels in pools 3 and 4 at roughly 16:45 also show up. The oscillation patterns as observed and as simulated are similar, although there are also differences. In reality, we do not know exactly how the real canal differs from the assumed canal. Our rough guess of what adjustments to make gave us the right kind of response, even though it did not match exactly.

In Figure 4, we show the results of SOBEK simulation for the conditions of October 16, 2001. In this case, our assumed conditions seem to match the observed conditions fairly well, although more oscillations were observed in the water level of pool 1 than what was simulated. This could also be due to gate hydraulic conditions that we are not accurately representing. By using untuned conditions (not shown), we found more oscillations in all pools under simulation than as observed, again except pool 1 late in the run.

DISCUSSION

We have demonstrated that the SACMAN control system is capable of controlling water levels in an irrigation canal. The basic components are working satisfactorily within a commercial SCADA package. The Automata hardware and firmware in the field is also performing as expected. Refinements are needed to make this system more failsafe so that it can run essentially unsupervised.

Initial tests with simple controllers suggest that simple PI controllers need to be very damped for this type of canal to avoid oscillations. These oscillations are caused by inaccurate estimates of the delay times for the pools that were used for controller tuning. Since canal pool properties change over time, one cannot expect to know these delays accurately. Much better control was observed for the PI^{+1}_{-1} controller that passed control actions to one additional gate upstream and one additional gate downstream. Further research needs to be done to document the performance of these controllers under a wider variety of conditions.

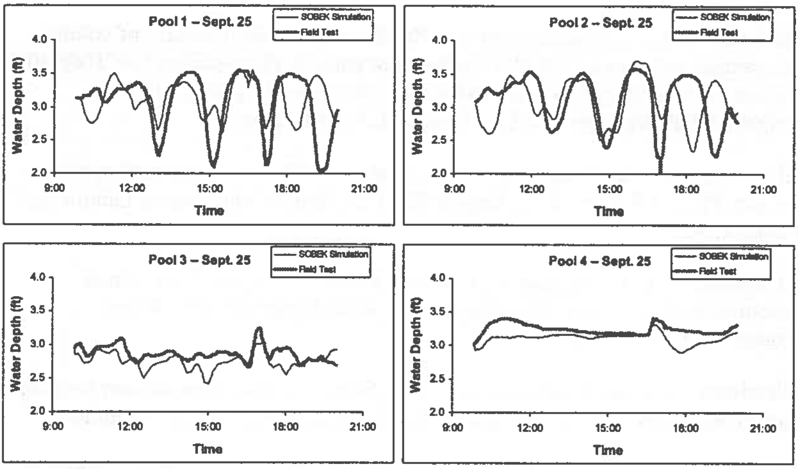

Figure 3. Field data and Sobek simulation results under untuned conditions for tests run on September 25, 2001.

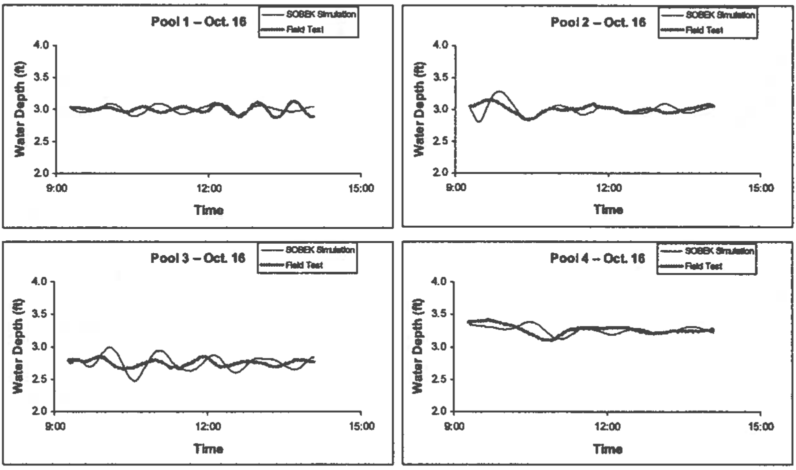

Figure 4. Field data and Sobek simulation results under tuned conditions for tests run on October 16, 2001.

REFERENCES

Bautista, E., A.J. Clemmens, and T.S. Strelkoff. 2002. Significance of volume compensation for open-channel feedforward control. Proceedings, USCID/EWRI Conference on Energy Climate and Water – Issues and Opportunities for Irrigation and Drainage, San Luis Obispo, CA 10-13 July.

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

Clemmens, A. J., E. Bautista, R. J. Strand, and B. T. Wahlin. 2001. Canal automation pilot project: Phase II Report. *WCL Report 24*. U.S. Water Conservation Lab., Phoenix, AZ

Clemmens, A. J. and J. Schuurmans. 2002. Simple optimal downstream feedback canal controllers: Theory. *Irrigation and Drainage Engineering*. Submitted.

Clemmens, A. J. and B. T. Wahlin. 2002. Simple optimal downstream feedback canal controllers: ASCE test case results. *Irrigation and Drainage Engineering*. Submitted.

Delft Hydraulics 2000. SOBEK flow module technical reference guide, version 2.2, Delft, The Netherlands.