TRAINING SCADA OPERATORS WITH REAL-TIME SIMULATION

R.J. Strand[1]
A.J. Clemmens[2]
N.T. Denny[3]

ABSTRACT

Many irrigation districts use SCADA software to manage their canal systems. Whether homegrown or commercial, these programs require a significant amount of training for new operators. While some SCADA operators are hired with extensive field experience, others are hired with no field experience at all and require extended training to gain an understanding of the behavior of open-channel systems. Additionally, many new operators have little or no experience with general computer use. Regardless of experience, these operators usually receive SCADA training while managing the actual canal system and their training is driven by the day-to-day operation of the system. Similar to the training needs of human operators, canal automation systems need to be tested for situations, such as hardware vandalism, a large storm event, or a canal breach, that cannot be created easily in an actual canal without wasting water, causing significant fluctuations in adjoining pools, or potentially causing severe damage. Additionally, the SCADA software itself needs to be tested for proper response to various alarm conditions. Currently, there are no practical methods to simulate emergency conditions for SCADA systems. The U.S. Water Conservation Laboratory (USWCL) has created a method for replacing the real canal with a simulation model without making any changes to the SCADA software. The connection from the SCADA computer to the radio is replaced with a connection to another computer that performs the canal simulation. If the simulation model were a good representation of the canal, gates, etc., the SCADA operators would not be able to tell the difference. This additional computer also runs software modeling the telemetry system, including communications, sensors and remote terminal units (RTU's) or programmable logic controllers (PLC's), as well as the physical components of the canal sites such as gate positions and battery voltages. Through the simulation system, situations such as a noisy transducer, a stuck gate, an electrical failure, or an unexpected supply or demand change can be simulated. If needed, these situations can be easily repeated. This paper describes this prototype system and potential uses for training and for validation of SCADA and automatic control functions.

---

[1] Electrical Engineer, USDA-ARS, U.S. Water Conservation Laboratory., 4331 E. Broadway Rd., Phoenix, AZ, 85040; bstrand@uswcl.ars.ag.gov
[2] Director, USDA-ARS, U.S. Water Conservation Laboratory., 4331 E. Broadway Rd., Phoenix, AZ, 85040; bclemmens@uswcl.ars.ag.gov
[3] Computer Programmer Senior, Maricopa Agricultural Center, University of Arizona, 37860 Smith-Enke Rd., Maricopa, AZ, 85239; denny@ece.arizona.edu

INTRODUCTION

USWCL Canal Automation Research Activities

The USWCL has been involved in canal automation research since 1991. Early research was focused on the use of canal simulation models to evaluate various canal automation methods. Since 1995, USWCL research has turned to the application of canal automation on existing systems and development of feedback and feedforward control methodologies.

In 1995, the USWCL started long term cooperative efforts with the Salt River Project (SRP) of Phoenix, Arizona. Originally, this work was an SRP pilot project to determine the feasibility of implementing automatic control on the main canals in the SRP delivery system and to test automatic control on a segment of one canal.

From 1995 to 2004, the USWCL was in a cooperative agreement with Automata, Inc. of Nevada City, CA, to develop a plug-and-play automation system using Automata hardware. This system is currently being tested on the WM-Lateral of the Maricopa-Stanfield Irrigation and Drainage District (MSIDD) in Maricopa, Arizona.
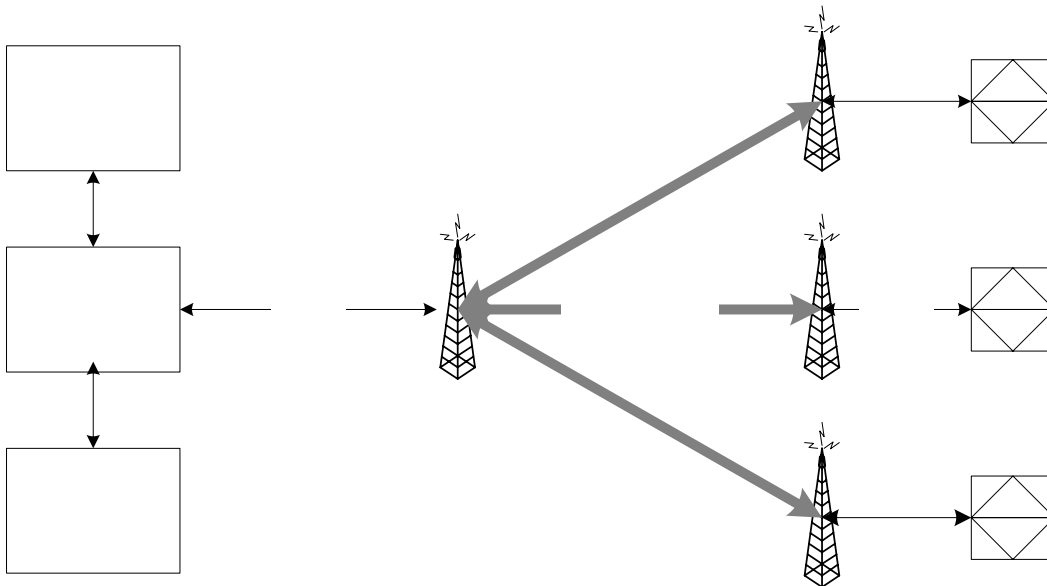

Figure 1. USWCL Canal Control System.

The USWCL canal automation system, shown in Figure 1, consists of various hardware and software components. The SCADA software is iFix by GE Fanuc

(formerly by Intellution). This software functions as a supervisory control interface and provides a data interface for automatic control. The USWCL has developed a suite of accompanying software programs, collectively called SACMan (System for Automating Canal Management) that implements local control, central feedback control, and feedforward control functionality and generates management information for use in the iFix supervisory interface (Clemmens et al. 2003). SACMan communicates with the iFix process database (PDB) through a proprietary library. The field hardware includes Automata pressure transducers, gate position sensors, and Automata's "Mini" RTU's. Communication between the iFix and the field hardware is over spread spectrum radios using the Modbus protocol on a serial RS-232 interface.

As part of the cooperative effort with Automata, the USWCL has assisted the Central Arizona Irrigation and Drainage District (CAIDD) in Eloy, Arizona with implementation of a district-wide hardware replacement and configuration of a new iFix SCADA installation in 2002. To date, 118 main and lateral canal check structures have been equipped with Automata RTU's, pressure transducers, and spread-spectrum radios. All of these devices are monitored and controlled in a supervisory fashion through a single iFix SCADA node.

Training Issues

While the USWCL has been involved with SRP, MSIDD, and CAIDD in canal automation research and implementation, a number of operator training issues have become apparent. First, operators start with varying levels of canal experience. At SRP, a very large organization, operators generally work their way through the hierarchy of the canal operations system. Most start as a zanjero and eventually work their way to the control room. Through this experience, they gain working knowledge of system topology, pool transmission delays, problem check structures, etc. This knowledge is invaluable once they are in the control room. While smaller organizations do promote some field operators to SCADA operator positions, these smaller districts are sometimes forced to hire SCADA operators with little or no canal operation experience.

Second, regardless of the level of applicable canal experience, the learning curve for a SCADA system can be quite steep, especially for an operator with limited computer experience. Some "homegrown" SCADA systems run essentially as embedded systems, meaning that the only software available on the computer is the SCADA software. With modern Windows-based SCADA systems, these computers can allow the operator to use other tools, such as email, databases, billing, and accounting software on the same computer. Many operators require some training to gain familiarity with a networked computer environment as well as managing critical processes in conjunction with other applications.

Finally, operators are trained in a "Live" environment while operating a real canal. At SRP, the environment is generally focused on running the canal system. There are separate departments that handle ordering, billing, and payment receipt. At MSIDD and CAIDD, the canal operators handle all of those financial/clerical tasks in addition to operating the canal. For those with little or no canal experience, the distraction of dealing with the business functions can be overwhelming and can hinder progress in understanding the system from an operational perspective.

Hydraulic Simulation Software

As previously noted, since 1991, the USWCL has made extensive use of hydraulic simulation models as part of ongoing canal automation research (Clemmens et al. 2005). These models have been used to determine canal pool properties and pool response for controller design and for validation of control methodologies. The models are useful because they essentially compress time by simulating hours of canal activity in a matter of minutes. One of these models, Sobek, by Delft Hydraulics, Delft, the Netherlands, has been used at the USWCL since the late 1990's. Originally released in 1993, Sobek uses an interface with the MATLAB computational environment to allow control decisions to be made within that framework (Delft Hydraulics 2004). Water levels are passed to MATLAB and gate positions are passed back to Sobek. The control routines are written as MATLAB "m" files (MathWorks 2005).

Simulators as Training Tools

Digital computer simulations have been used for decades by the military to train fighter jet pilots in flight simulators. Additionally, flight simulator time has become part of required yearly training for commercial airline pilots. Digital simulations have also been employed in medicine. Today, surgeons are able to practice optical and endoscopic surgery with visual feedback generated by digital models. In parallel to these advanced uses of digital simulations, the USWCL has harnessed the capabilities of Sobek to create a simulation based training tool for canal operators.

APPROACH

While conceptualizing the simulation system, a number of requirements became clear. First, the SCADA system configuration had to remain intact. Even in moderately sized implementations, the configuration of SCADA software can be time consuming and complex. To avoid reconfiguring an established system, the simulator had to interface with the SCADA system at the external communication layer – the same layer used to communicate with the RTU's or PLC's. In the case of the USWCL system, this meant that the simulator had to use the Modbus

protocol to exchange sensor information and gate movement commands with the SCADA system.

By moving the communication to this layer, switching from the real canal to the simulator is accomplished by simply unplugging the serial cable leading to the radio and replacing it with a serial connection to another computer. Additionally, use of the external communication layer allows simulation of errant transmissions and transmission delays, as well as the implementation of other serial and network communication protocols. Finally, moving the communication to the external layer allows the simulator to be used with any SCADA system using Modbus, not just iFix.

Second, the simulator had to provide a good representation of the field hardware. This meant that the simulator had to replicate the embedded programming of the Automata RTU, A-D conversion, and sensor types and ranges. This allows testing of various types of problems ranging from sensor noise to gate motor failure.

As shown in Figure 2, the simulator software is implemented as two processes. One process, the physical simulator, maintains physical information about the canal such as gate positions, water levels, battery voltages, and turnout flows. The physical simulator has an interface which allows a user to implement field changes such as adjusting a turnout flow or starting a source well.
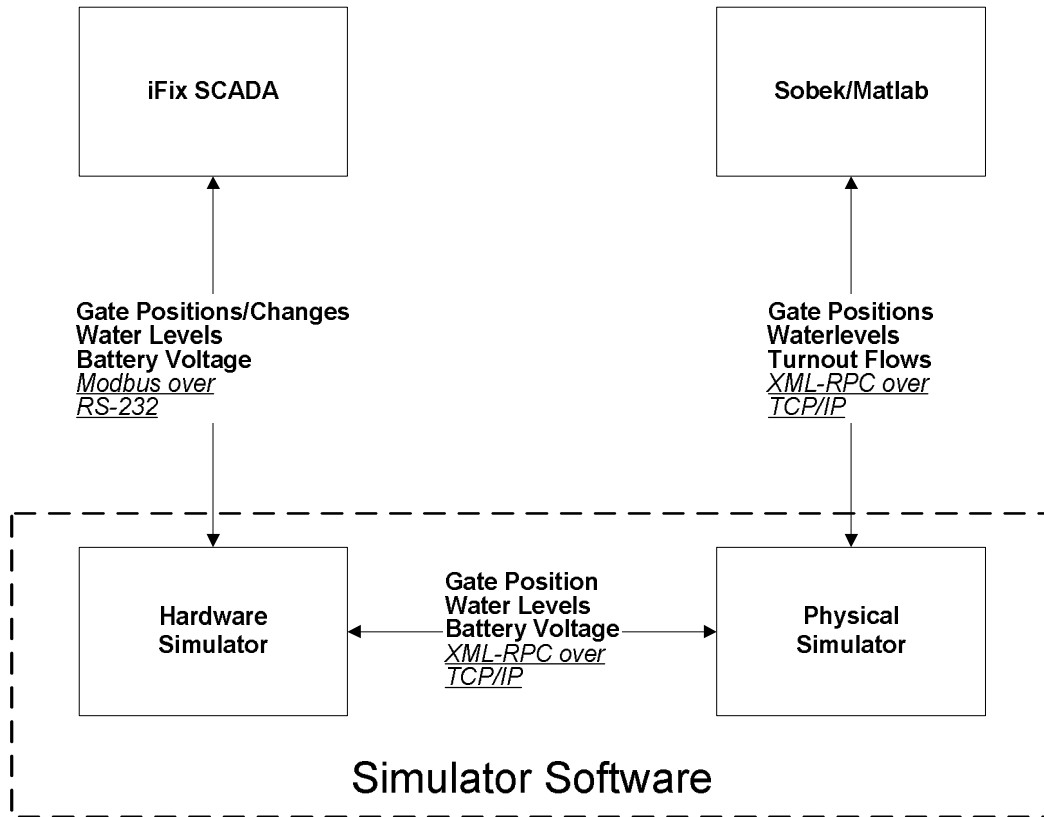
Figure 2. Simulator Components and Communication.

The second process, the hardware simulator, emulates the RTU, its associated sensors and relays, and the gate motor. Perturbations such as noise or a stuck gate are introduced in this process and transmitted to the SCADA system. In order to implement movement delays and sensor warm-up and to accommodate the implementation of real-time embedded process in the field hardware, this process actually creates a separate thread to emulate each RTU. A user interface allows modification of delays and addition of noise to a gate position or water level.

The processes were separated for a number of reasons. First, the properties maintained in the physical simulator are considered to be actual real-world values. To maintain a stable Sobek simulation, these values are not modified by noise or sensor malfunction. Second, separation of the processes also facilitates independent communication rates with MATLAB and the SCADA system.

As previously stated, in normal use, Sobek computes canal responses very quickly. For use in a real-time simulator, each Sobek timestep is delayed until real-time has caught up to the simulation time. This is accomplished by implementing delays in the MATLAB "m" file.

A number of operational sequences take place to move information between the SCADA system and Sobek

Interrogation of a pressure transducer for a water level requires the following steps:
- Whether initiated by a SCADA user query or routine timed polling, the SCADA software sends a Modbus signal over the serial cable to the hardware simulator.
- The hardware simulator parses the Modbus command and initiates a sensor reading by the appropriate RTU thread
- The RTU thread gets the most recent water level from the physical simulator, emulates the A-D conversion and adds any noise prescribed though the hardware simulator interface.
- The result is sent back to the SCADA system in the Modbus reply message.

Moving a gate requires the following steps:
- The SCADA operator implements the gate movement through the SCADA interface.
- The SCADA software sends a Modbus signal over the serial cable to the hardware simulator.
- The hardware simulator queries the physical simulator for the current gate position, calculates the new gate position and then writes the updated gate position to the physical simulator.
- On the next Sobek timestep, the physical simulator passes the new gate position to MATLAB which then sends it to Sobek for future calculations.

In order for Sobek to update a water level, the following steps take place:
- Sobek calls the MATLAB routine
- MATLAB examines the timestamp of the Sobek data. Generally, Sobek is ahead of real time, so MATLAB waits in a delay loop until real time catches up to the Sobek time.
- MATLAB passes the water level to the physical simulator.
- This water level is then available for the hardware simulator.

Field events, such as a turnout change, require interaction with the physical simulator through a user interface.
- The user enters the new turnout flow in the physical simulator user interface.
- On the next Sobek timestep, MATLAB recognizes the flow change and calculates a new gate opening based on either the current water level or a setpoint water level.
- This new turnout gate position is then passed to Sobek for use in subsequent calculations.

The Extensible Markup Language (XML) file format provides a method for combining data and descriptive information for that data in a single text file. XML-RPC (XML Remote Procedure Call), an extension of XML, provides a method for implementing cross-platform distributed computing. XML-RPC processes communicate using standard Internet protocols. Through XML-RPC, a process on one machine can call another process, on the same machine or on another networked computer, and pass XML encrypted data in the function call.

As shown in Figure 2, the two processes making up the simulator use XML-RPC to communicate with each other as well as MATLAB. Each process has a front-end which essentially functions as a web server. This front end processes XML-RPC calls from the other sources. By using a network based communication protocol, the processes can be distributed among multiple networked computers in very large implementations. MATLAB and Sobek can run on one computer, the physical and hardware simulators can each run on separate computers, and the SCADA computer remains untouched. The communication between the hardware simulation and the SCADA system uses the Modbus protocol over an RS-232 serial connection.

USE AS A TRAINING TOOL

As this new functionality was being developed, it became quite clear that while this system was going to be of great benefit for the internal testing of the USWCL control system, the simulator would also be an excellent training tool.

Any irrigation district that uses a computerized control system should maintain a duplicate system as a backup in case of a failure of the primary system. This backup system could also be used as a training system for operators. Sobek, MATLAB, and the simulator software can run on a separate computer representing the "Canal Side" of the system.

One advantage of using the simulator for training is that the trainee can be isolated from the distractions of the control room and can focus on computer skills and canal management. Another advantage is the ability to repeat a given scenario. This is almost impossible to accomplish while training with the real canal.

Training Scenarios

One of the first training sessions could be simply routing ordered deliveries through the simulated system. This could initially be done in a supervised manner and then repeated by the trainee. The results could be compared by reviewing the historical data from the SCADA system.

Alarm situations could be the next scenario. Each organization has allowable tolerances for various measurements and procedures for dealing with related alarms. Scenarios can be created to simulate high, low or quickly changing water levels, sensor failure, communication failure, or intermittent sensor spikes.

One of the most common situations requiring operator intervention is an unexpected offtake change. Whether caused by an unexpected shutdown at a water treatment plant, power malfunction at a supply well, or high winds at a time when grain is tall and headed out, these emergency changes can require intense effort on the part of the supervisory SCADA operator. Again, the ability to replicate and review individual scenarios could be of great use in training a new operator.

Although less frequent, gate malfunctions are a problem situation. Whether due to long-term use or vandalism, these malfunctions can cause huge water level deviations and can require intense effort on the part of the SCADA operator to avoid damage to the canal. In the simulator, it is quite simple to force a gate to close and stop responding to control commands.

This simulator is also valuable for training with SCADA systems utilizing automatic control. Even with field experience, operators face a learning curve associated with operation and configuration of the control system as well as understanding the actions taken by automatic control methods. Additionally, operators need to understand the limits of the automatic control system and learn when to intervene and switch to supervisory control.

FUTURE ENHANCEMENTS

As previously stated, the hardware simulator only supports the Automata "Mini" RTU as it is configured for use in the USWCL canal automation system. It would be desirable to expand this to other hardware. One of the issues is in replication of the embedded functionality of these devices. Assimilation of the various forms of ladder logic, assembly language, and other programming languages combined with hardware specific addressing makes it difficult to simply import the program code for another programmable logic controller (PLC) or RTU. Recent work by the International Electrotechnical Commission (IEC) to develop a PLC programming standard may make this integration feasible in the future.

Similarly, the hardware simulator can be upgraded to handle various publicly available communication protocols in addition to Modbus.

In addition to Sobek, the USWCL also has extensive experience with the MIKE11 simulation model from the Danish Hydraulics Institute (Danish Hydraulics Institute 1995). The user interface for this program is through code modules compiled in Borland Delphi. It would be useful to create an XML-RPC library

for use in Delphi, thereby allowing MIKE11 to be used to generate the canal response for the simulator.

CONCLUSION

For research purposes, this simulator has already been a valuable tool for debugging the USWCL control system in preparation for field trials at MSIDD. In its current form, the simulator is ready to be used as a training tool with Modbus based SCADA systems. With some enhancements, the simulator could be used with other communication protocols and could simulate a wide range of field hardware.

REFERENCES

Clemmens, A. J., Strand, R. J., and Feuer, L. 2003. Application of canal automation in central Arizona. p. 455-464. In 2nd Int. Conference on Irrig. & Drain, Phoenix, AZ, May 12-15, 2003.

Clemmens, A. J., Bautista, E., Wahlin, B. T., and Strand, R. J. 2005. Simulation of automatic canal control systems. Journal of Irrigation and Drainage Engineering 131(4):324-335.

Danish Hydraulics Institute (DHI). 1995. Mike11 Version 3.11 user manual, Horsholm, Denmark

Delft Hydraulics 2004. Technical reference for real time control, Version. 2.09.

MathWorks 2005. MATLAB user's guide. Version 7. The Math Works Inc., Natick, MA.