REAL-TIME MODEL-BASED DAM AUTOMATION: A CASE STUDY OF THE PIUTE DAM

Matthew S. Maxwell[1]

ABSTRACT

For over five years, Piute Dam in the Sevier River Basin of Utah has been equipped with an automatic gate to regulate releases into the downstream irrigation delivery system. This system has allowed the water commissioner to remotely set the desired flow target. The gate then automatically adjusts itself to provide the specified outflow. Although this has been very convenient, the determination and setting of the target release flow rate has remained a human-initiated action. Recent improvements to the system have now automated this part of the process to provide a greater degree of convenience and efficiency.

Currently, Piute Dam is being used as a test bed for automation technology. In conjunction with the Sevier River Water Users Association (SRWUA), the Bureau of Reclamation's Provo Area Office developed software to enable automation of the Piute Dam outlet gate. Software used in this process includes the OpenBasin software package for real-time data acquisition, a model developed for Piute Reservoir by Abedalrazq Khalil and Mac McKee of Utah State University, and other software used to develop Internet-based control and reporting. Furthermore, the software developed to provide supervisory control, policy enforcement, and diagnostics was integrated into the OpenBasin software package. This software is open-source and available for free. This allows other water districts to apply these tools to their own automation projects.

INTRODUCTION

The successful implementation of a real-time model-based dam automation system is an important step toward developing a convenient and efficient water management strategy. Furthermore, dam automation is an important hurdle to overcome on the path towards a fully automated river basin. Currently, Piute Dam in the Sevier River Basin of Utah is being developed as a test bed for automation technology. This technology is designed to collect real-time data from sensors along the river, use these data to calculate a desired reservoir release according to a hydrologic model, and autonomously adjust the outlet gate on the dam to reach the desired release.

For the purposes of this paper, 'dam automation' will refer the process of changing the reservoir release gate to accommodate a change in downstream demand without human intervention. The two terms 'model-based' and 'model-

[1] Computer Assistant, U.S. Dept. of the Interior, Bureau of Reclamation, Provo Area Office, 302 East 1860 South, Provo, UT 84606-7317

driven' indicate that the desired reservoir release is dictated by a hydrologic model and will be used interchangeably.

This article will present a brief overview of the Sevier River Basin and the technology implemented there, describe the model-driven automation process from data acquisition to remotely controlling the reservoir's release gate, highlight supervisory controls built into the automation software, and discuss future developments in the Piute Dam automation software.

SYSTEM DESCRIPTION

Sevier River Basin

The Sevier River Basin is a closed basin located in rural south-central Utah. It covers approximately 12.5% of the state of Utah and is managed by the Sevier River Water Users Association (SRWUA). The majority of water in the basin is used for irrigation purposes. The basin is divided into five regions as represented in Figure 1: Upper, Central, Gunnison, Lower, and San Pitch. Piute Reservoir is located in the Upper region. Valuable run-off collects in Piute Reservoir each spring. Releases from this reservoir flow down into diversions located along the river in the Central region. Excess water runs over the Vermillion Dam (near the border of the Central and Gunnison regions) and is lost to water users in the Central and Upper regions.

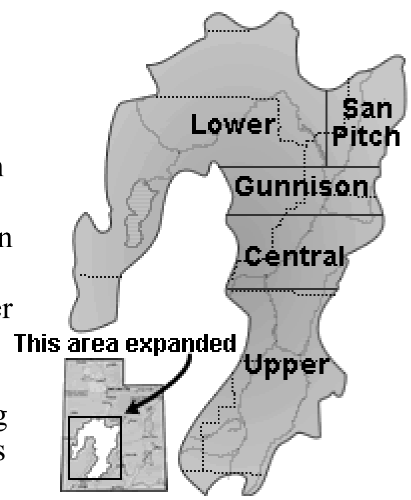

Figure 1. Sevier River Basin Map.

Because the Sevier River Basin is located in an arid climate and generally uses all possible water resources each year, effective management of Piute Reservoir is a high priority. To facilitate a more efficient water management system, the Sevier River Basin has been heavily instrumented and has many diversions and release structures that can be controlled remotely via the Internet. The gate controlling releases from Piute Dam is one of these remotely automated structures. This automated gate is a key component of the Piute Dam model-based automation project.

Technology Configuration

Three servers form the core of the Sevier River Basin's technology implementation: the data collection server, the model server, and the database server. The reason for these three servers is twofold: to accommodate existing technologies and to balance the workload. The data collection software and the modeling software both require extensive resources to perform properly and in a timely manner. To better facilitate resource utilization, these two applications are placed on different servers. Both of these applications require the Windows operating system; however, the database and website generation software runs under the Linux operation system. For this reason the database and website software must also reside on its own server. These three servers each have their own specialized responsibility and communicate with each other by sharing data through networked directories.

The data collection server is responsible for communicating with the remote stations via radio and for collecting data from these stations every hour. This server is also responsible for communicating desired reservoir releases to the remote terminal unit (RTU). The software that interfaces with the radio communications is the LoggerNet software package by Campbell Scientific, Inc. Not only does this software provide a graphical interface to interact with remote stations, but an additional software development kit (SDK) can be purchased to facilitate automation. Since LoggerNet only works on Windows machines, this server is running Windows XP.

The model server is a high performance machine dedicated to running complex models in a timely manner. This server is configured with all of the software needed by the hydrologic model. In this instance, the MATLAB software package by MathWorks was needed to run the hydrologic model developed by Utah State University. Due to model requirements, this server also runs Windows XP.

The database server is responsible for storing the data collected from remote stations by the data collection server. In addition to storing historical data in a database, this server hosts the SRWUA web page (http://www.sevierriver.org) and the web-based controls for dam automation. Data storage and dynamic website administration is achieved using the OpenBasin software package developed by the Bureau of Reclamation’s Provo Area Office and StoneFly Technology. Due to requirements of both the database and the OpenBasin software package, this server runs the Linux operating system. It is also the main server in the automation process and orchestrates the interaction of the other servers.

The RTUs used in the Sevier River Basin are Campbell Scientific CR-10x dataloggers. In addition to recording battery voltage, water height, gate height, and calculating water flow, many of these RTUs are connected to automatic gates and programmed to allow automatic gate adjustments. One such programmed feature allows a user to input a desired flow into a storage register in the datalogger. The RTU will then automatically move the gate until it reaches the flow and further adjust the gate to maintain the flow. The dam automation software uses this feature to automatically input model-based reservoir releases into the datalogger at Piute Dam. From there, the software simply allows the RTU to adjust the gate accordingly.

The interconnection between these servers and the dataloggers is illustrated in Figure 2.

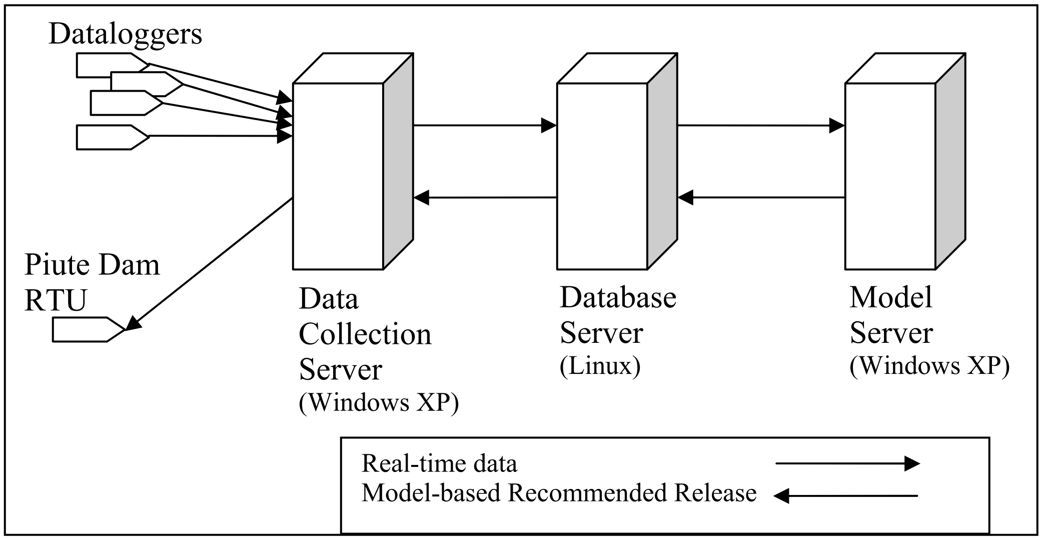

Figure 2. Flowchart of Technology Interconnections.

Model Description

The hydrologic model used for this dam automation project was developed by Abedalrazq Khalil and Mac McKee of Utah State University. The model is a Bayesian adaptive learning model that is not based on physical models of the basin; instead, this statistics-based model is able to detect changes in water demand and compensate for those changes. Relevance vector machines (RVMs) were used to detect abnormalities and drift in the system (Khalil and McKee, 2004). The input to the model includes real-time data from downstream stations and the current reservoir release. The output of the model is the recommended release from the reservoir based on the statistical analysis of river basin conditions and previous data. This recommended release is sent to the RTU that controls the Piute Dam gate.

AUTOMATION PROCESS

The purpose of the automation project is to regulate the reservoir release to a specified amount as recommended by a hydrologic model. This is done without human-intervention. Recommended releases are calculated by the model every hour and are immediately applied to the dam release gate. Furthermore, bounds are set on the automation process to prevent erroneous behavior, and the model can be overridden through human interaction at any time.

There are three main steps in the dam automation process. First, real-time data used by the model must be collected and stored. Second, these data must be inserted into the model. Third, the model results must be automatically applied to the gate on Piute Dam. Furthermore, in the Sevier River Basin, this process is complicated by the fact that the data collection software, database, and model all reside on different computers. Through a series of shared directories between the computers and specialized software applications on each server, the OpenBasin software is able to communicate with the other computers and complete the automation process.

Data Acquisition

Communication with the remote stations is performed entirely by the Campbell Scientific LoggerNet software. This software is configured to automatically radio to each remote datalogger, collect the associated data, and store it to a text file each hour. The directory in which these data files are stored is accessible from the database server. The OpenBasin software, running on the database server, checks the data files on the data collection server every 10 minutes and inserts new records into the OpenBasin database.

System/Model Integration

Once OpenBasin has detected that it has all of the required information to run the model, OpenBasin will assemble a text file used as input for the model. OpenBasin will drop this file into a shared directory on the model server. In order for the model to run, we developed a small software application called the Model Server Monitor that runs on the model server. This application checks for the presence of the model's input file. Once this file is detected, it will automatically run the model using the file as input for the model. The model usually takes about 5 minutes to finish; however, if the model detects novelty in the system (i.e. the model needs to adapt to new conditions), it can take up to 30 minutes before completion. The results of the model are stored in an output file. Once the OpenBasin software detects the presence of this output file, it parses the file and stores the model's output into the OpenBasin database. After the data are successfully stored in the database, both the input file and the output file are deleted to allow the process to continue correctly on the next iteration (i.e. the next hour when the entire process is repeated).

Automating the Dam

After the model's results are stored in the database, OpenBasin begins the process of automatically adjusting the gate to achieve the desired reservoir release. This is done by generating another input file and storing it in a specific directory on the data collection server. Like the model server, the data collection server has a software application called the Datalogger Server Monitor that checks for the presence of this input file. However, unlike the Model Server Monitor, which immediately runs the model after detecting the input file, the Datalogger Server Monitor has a specific window of time within which it can make radio connections. This window of time is approximately 10 minutes before the next hour (data collection for the remote stations begins on the hour). The Datalogger

Server Monitor will wait until it enters this window before processing the input file and sending the target flow rate to the RTU. The communications to the datalogger are performed through the LoggerNet software and automated using the LoggerNet SDK.

After the parameters are correctly set on the RTU, the RTU will automatically adjust the gate height to reach and maintain the desired release. The flow through the gate is measured by a remote station located just below the dam. Periodically, this value is communicated to the RTU at the dam via radio and a PI feedback controller is used to determine gate movements required to regulate the flow. Through the same process used by the Model Server Monitor, the Datalogger Server Monitor stores its output into a file, and OpenBasin parses the file and stores the results into the database. As the final step in the automation process, a diagnostics report is made to document the model/automation status. From start to finish, the entire process takes nearly one hour (including waiting time), and it begins again as soon as new data is available.

SUPERVISORY CONTROLS

Throughout the whole automation process, errors may occur. Errors in the data acquisition process may cause erroneous data or missing data. Imperfections in the model may recommend impractical or incorrect reservoir releases. Additionally, the user must be able to turn the model-based automation on and off easily and override any release recommended by the model. For these reasons, the dam automation process must be robustly implemented in order to successfully handle errors, and failsafe mechanisms must be implemented to ensure that failure in the dam automation process will not adversely affect the actual reservoir.

The supervisory controls developed for this automation process allow a user to specify a range of normal behaviors and dictate corrective actions in advance. Consequently, an hour-by-hour approval for the dam release is not necessary. The model-based automation will proceed without human intervention until a detectable error has occurred. At that point the process will take corrective action as determined in advance and alert the user. Depending on the nature of the error and the corrective action configured, the automation process may be able to continue correctly without human intervention at all.

Policy Enforcement

To help ensure that the dam automation works as desired, a number of policies were hard-coded into the automation program. The policies are defined by configuring a corrective action to take when a specified error has occurred. The first step toward enforcing these policies is the detection of abnormalities. Examples of abnormal behavior include prolonged absence of real-time data, model errors, recommended releases surpassing predetermined minimum and maximum values, and too much variation of reservoir release values. These abnormalities are indications of either erroneous data collection or a malfunctioning model. Detection of abnormalities happens automatically within the dam automation software running on the Database Server.

When abnormalities are detected, an associated corrective action is taken. Example actions include ignoring the recommended release for the given time step, automatically shutting down the model-based automation to prevent further abnormalities, and falling back to a "safe" reservoir release as determined by the user. Additionally, the policies may be configured so that the model-based automation may automatically turn itself back on if the model's estimates return to normal. These corrective actions are all determined beforehand and happen without intervention. When corrective actions are taken, the system alerts the user and reports that it has encountered abnormal behavior so that the user can further decide how to respond.

Each type of abnormality can have a grace period associated with it to prevent actions from taking effect until a number of consecutive abnormalities are detected. For example, the policies can be configured so that the model will ignore any reservoir release above or below the set minimum and maximum values and automatically shut down the model and return to a "safe" reservoir release if the model continues to request abnormal releases for five continuous time steps. The parameters used to detect abnormalities, associate corrective actions, and set grace periods are all configured through the web-based control panel.

Web-based Control Features

The web-based control panel provides the user with supervisory control over the model-based automation process. From the website, the user can turn the model-based automation on and off, override the model's recommended release, and configure policy enforcement parameters. The existence of a web-based control panel allows the user access to these features from any Internet-enabled computer. For improved security, this website is password-protected so that intruders may not maliciously affect the reservoir release.

Additionally, as an emergency shutdown mechanism, the user can connect to the RTU with the LoggerNet software and turn the model off completely by setting an appropriate value on the datalogger. The software designed to automate the reservoir release checks this value before making any automatic gate changes. This manual shutdown may be used when the web-based control panel is unavailable or in any other emergency situation.

Diagnostic Information

The dam automation software meticulously records the status of its operation and generates alarms, notifications, and status reports based on these records. Each time the model runs, it checks for erroneous data, model error, and other abnormal behavior. If any of these errors occur, the type of error encountered is stored in the database. Additionally, if the dam automation software successfully runs, it stores the reservoir release amount into the database as well. With this information, one is able to see an hour-by-hour view of exactly what is occurring with the dam automation.

Since a complete hour-by-hour view may be too complex to easily decipher, the software aggregates these data and displays it via the Internet in different levels of detail. Low-detail views can be used to get a general idea of how the model-based automation is currently working while high-detail views can be used for troubleshooting errors.

This diagnostic information is deployed using the Really Simple Syndication (RSS) protocol. The use of a standard protocol allows a greater degree of interoperability between software. For example, free programs called RSS readers can be installed on a user's computer. These programs can be configured to check the diagnostic information for the dam automation software. Every time the diagnostic information updates, the RSS reader will gather the new diagnostic information and load it onto the user's computer. This ensures that the current diagnostic information is easily accessible by all who need to access it. For even greater accessibility, these RSS files are displayed on the web-based control panel and on other diagnostic web pages.

FUTURE DEVELOPMENTS

Although functional, the current set of technologies used to implement model-based automation of Piute Dam could be improved in many ways. First, the current software package is very dependent on the setup of the SRWUA systems and may require a certain degree of customization to transfer the software to another system. Developing the software with network programming techniques would make inter-computer communications more reliable and eliminate the platform-dependence of the software. Second, building upon the current diagnostic tools, an alarming system that can communicate via email or phone would be a useful addition for times of emergency or uncertainty. Third, the accessibility of the web-based control panel could also be improved by designing a special version of the control panel for web-enabled cell phones.

Network Programming

The current software used to automate Piute Dam is very dependent on the network setup of the Sevier River Basin. Specifically, the use of shared directories between servers may not always be possible in the case of hydrologic models hosted offsite. Additionally, the Model Server Monitor and the Datalogger Server Monitor only run on the Windows XP operating system. Through the use of network programming techniques, the automation software, model software, and datalogger software could all communicate in the same way computers communicate through the Internet. This would remove all platform dependencies of the software and enable the automation software to be more easily implemented on systems with different configurations.

Email/Phone Alarms

The current system of diagnostics and alarms provides a great way to display information for those who monitor the automation software's status. However, during times of extreme abnormality it would be useful for the program to pro-actively contact the user. Two ways of facilitating this are email- and phone-based alerts. Email-based alerts could be readily added, but they do not assure immediate communications to the same degree that phone-based alerts would. Fortunately, properly configured policies in the automation software should gracefully handle errors without immediate intervention. Alarms via email or cell phones would be extremely useful if integrated into an alarming system for the entire river basin's technology implementation. Current development on this system is underway.

Cell Phone Supervisory Control

The job of a water commissioner is often one requiring much time out of the office and away from technology. One of the current features of the OpenBasin software package is its ability to display the real-time status of river basins on special text-based web pages for cell phones. In addition to this feature, the development of a system that would allow supervisory control functionality via a web-enabled cell phone would greatly increase the accessibility and usefulness of the model-based dam automation software. Additionally, the control panel could be extended to provide supervisory control of more parts of the river basin via the Internet or a web-enabled cell phone.

Software Integration

After the software described in this paper is refined and enhanced, it will be bundled with the OpenBasin software package. The OpenBasin software is used to collect, store, manipulate, and display data easily via the Internet. It is available for free on the OpenBasin website (http://www.openbasin.org). After these additions are made, the dam automation software will be immediately available to various river basins throughout the state of Utah that use this software and have similar automation technology installed.

CONCLUSION

The techniques and software used to implement real-time model-based automation at Piute Dam have proven to be very effective. The process undertaken to accomplish the dam automation project has been described from data collection to automation of the release gate. Additionally, a description of the technological setup has been given. Future improvements upon the automation software’s usefulness, features, and techniques are still being actively developed and refined. After finishing a more unified version of the technologies described in this paper, the software will be bundled with the OpenBasin software package. Additionally, the software used in this automation project will be applied to other water districts within the state of Utah in the near future and improved upon each step of the way.

REFERENCES

Khalil, A. and M. McKee. 2004. Hierarchical Bayesian Analysis and Statistical Learning Theory II: Water Management Application. 2004 USCID Water Management Conference Proceedings, 445-455.