

MODEL-BASED AUTONOMOUS CONTROL OF PIUTE DAM

by

Matthew S. Maxwell

Submitted to Brigham Young University in partial fulfillment
of graduation requirements for University Honors

Computer Science Department

Brigham Young University

April 2006

Advisor: Sean Warnick

Honors Representative: Craig Coleman

Signature:_____ Signature:_____

ABSTRACT

MODEL-BASED AUTONOMOUS CONTROL OF PIUTE DAM

Matthew S. Maxwell

Computer Science Department

Bachelor of Science

This thesis deals with the control-oriented multiple input single output (MISO) system identification process for Piute Dam and Sevier River. The Sevier River is an open-channel river with long delays and infrequent data measurements. Based upon adaptations of two models commonly used to estimate river flow, we developed a parameterized mass balance model and compared the validation results of the three models. The parameterized mass balance model performed the best during periods of high flow. Since high flow conditions are most important for water conservation efforts, we chose this model as the basis of a robust controller to be designed and implemented on the Sevier River Basin.

Using the models obtained from the system identification process, we build a model of the entire system and use this as the basis of our controller. We use integral control to ensure asymptotic tracking of a reference signal and use optimal gain measurements to calculate appropriate reservoir releases. Furthermore, we adapt the

integral control to work properly with the heavy delay of the system. We verify the accuracy of the controller by using validation data in a closed-loop simulation.

This controller will be implemented on the Piute Dam in the spring of 2006 in an attempt to provide technological methods to improve water management convenience and efficiency. Detailed information concerning the technology configuration of Sevier River Water Users Association and the automation framework for the real-time model-based automation project are also included in this thesis.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the assistance of my advisors Sean Warnick of the Brigham Young University Computer Science Department and Roger Hansen of the U.S. Department of Interior, Bureau of Reclamation's Provo Area Office for providing many insights, clarifications, and resources to further this water management research. Jeffrey Humphreys of the Brigham Young University Mathematics Department was also instrumental in providing mathematical and technical expertise. I acknowledge the Sevier River Water Users Association's willingness in providing historical data and allowing the automation project to be implemented on Piute Dam.

This research was supported by the U.S. Department of Interior, Bureau of Reclamation and the Brigham Young University Office of Research and Creative Activities.

TABLE OF CONTENTS

Title and signature page	i
Abstract	ii
Acknowledgments.....	iv
Table of Contents.....	v
List of Tables and Figures.....	vi
 I. Introduction	 1
II. System Description	3
III. Modeling Procedures	8
IV. Modeling Results and Analysis	14
V. Control Design	18
VI. Implementation	23
VII. Conclusion.....	32
References	33
Appendix A: Transfer Functions	34

LIST OF FIGURES

FIGURE 1: Sevier River Basin Map	3
FIGURE 2: Piute Reservoir to Vermillion Dam Diagram.....	4
FIGURE 3: Flowchart of Technology Interconnections.....	8
FIGURE 4: Validation of Mass Balance Model	14
FIGURE 5: Validation of Second-Order Model with Delay	15
FIGURE 6: Validation of Parameterized Mass Balance Model	16
FIGURE 7: Validation of System Model.....	18
FIGURE 8: Sample Feedback System.....	19
FIGURE 9: Step Response of Feedback System for $k_i = .0275$	20
FIGURE 10: Step Response of Feed-forward System.....	21
FIGURE 11: Block Diagram of Delayed-Integral Controller.....	21
FIGURE 12: Step Response of Final Controller.....	22
FIGURE 13: Validation of Controller	23
FIGURE 14: Screenshot of Online Supervisory Control Features	28

LIST OF TABLES

TABLE 1: Ranges of System Conditions	16
TABLE 2: Model Validation Errors	17

I. INTRODUCTION

The conservation of water resources has become an increasingly important task throughout the world. This is especially true in arid climates where lack of water can cause economic ruin or struggle for survival. However, from a water management perspective there are many complexities that hinder water conservation efforts. Examples of these complexities include the size of river systems, the difficulty of accurately predicting downstream flow in the presence of disturbances, and the inability of water management professionals to constantly and consistently deliver water to specified locations on demand without incurring losses.

With the modernization of the water management industry and the rise of technological advancements enabling real-time water monitoring at remote locations along rivers and canals, there is a large potential for decreasing water losses through modeling and automatic control of river systems. There has already been much research dealing both with the modeling and the control of irrigation canals. An overview of current methods, applications, and research of this area is covered in [1].

Commonly, the modeling of open-channel hydraulics such as irrigation channels has been based on the Saint-Venant equations. These non-linear hyperbolic partial differential equations can accurately approximate the flow in such systems. Litrico, et al. have done an extensive amount of work with these equations and have developed suitable models for control design using simplifying assumptions and linearizing around a reference flow. The resulting system was able to be approximated using a second-order transfer function with delay [2]. With this model, Litrico, et al. were able to develop a

number of different controllers for the system including a PI controller [3], a robust IMC controller [4], and an H_∞ controller [5].

Additionally, Weyer, et al. have used data-driven system identification techniques to develop suitable models for irrigations systems [6]. In [7], Weyer, et al. show that data-driven models perform comparably with models derived from the physics-based Saint-Venant equations. Consequently, Weyer, et al. have created a number of different controllers for the data-driven model including an LQ [8] controller, an H_∞ controller [9] and others.

Although Weyer, et al. found suitable data-driven models for their system, these results do not scale well to the Sevier River Basin. This is due to the fact that the system modeled by Weyer, et al. had relatively small delays (a few minutes) and frequent data sampling (every minute). Because of the remote locations and battery-operated equipment in many areas of the Sevier River Basin only hourly data transmission is feasible. Furthermore, the lags in the Sevier River are usually over a day long.

The purpose of this thesis is to develop a data-driven model suitable for heavily lagged systems with infrequent data. Once this model is completed, we will design a feedback controller for the Piute Reservoir. After completing the controller we will implement the controller on the Piute Reservoir in the spring of 2006.

First, we provide a description of the system to be modeled including the physical layout, data collection and dissemination procedures, and technology configuration. Next, we discuss our modeling procedures including system representation, lag determination, and model classes used. We next show the results of the respective models and provide some analysis on the results. After assembling a model for the entire system, we will

build a feedback controller based around that model and verify the quality of that controller. Following that discussion we will cover implementation issues including the automation process and supervisory controls. Last, we present our conclusion and detail plans for future improvements and modifications.

II. SYSTEM DESCRIPTION

A. Physical Layout

The Sevier River Basin is located in rural south-central Utah. It covers approximately 12.5 percent 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. The Piute Reservoir is located in the Upper region. Valuable run-off collects in the Piute Reservoir each spring. Releases from this reservoir flow north into diversions located along the river in the Central region. Excess water runs over the Vermillion Dam (located on the Central/Gunnison border) and is lost to water users in the Central and Upper regions.

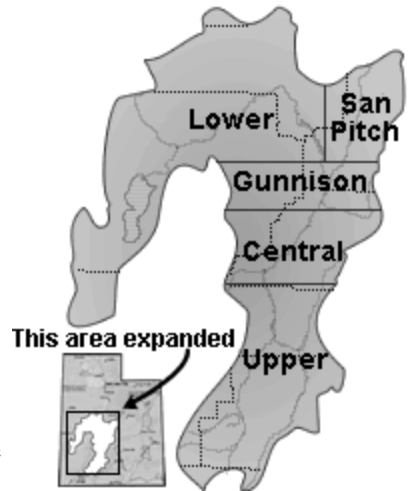


Figure 1. Sevier River Basin Map

Because the Sevier River Basin is located in an arid climate and generally uses all possible water resources each year, effective management of the 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/release structures that can be controlled remotely via radio. The Piute Dam is one of these remotely automated

structures. This automated gate forms the framework from which the Piute Dam model-driven automation project is based.

Average delays from the reservoir release at Piute Dam to the lowest downstream point in the model, Sevier River at Vermillion, are between 24 and 36 hours. The release from Piute Reservoir ranges from 0 m³/s in the winter months to nearly 20 m³/s at the height of the irrigation season. The average release during the irrigation season is around 8 or 9 m³/s.

B. Data Collection

Data are collected by remote data logging stations every hour. Measurements including water height, flow, etc. are transmitted via radio, phone, and/or internet to a central datahut. Each of these measurements represents averages over the previous hour.

Data collection software collects the data from each of the remote stations and stores them into a database. The locations of data collection sites along the Piute Reservoir stretch of the river are indicated by black dots in Figure 2.

As can be seen in Figure 2, there are two intermediate measuring stations located on the Sevier River: one above Clear Creek and the other near Elsinore. These intermediate measurements split the river system into three stretches as denoted by dashed lines on the graphic. Each of these three stretches can be modeled

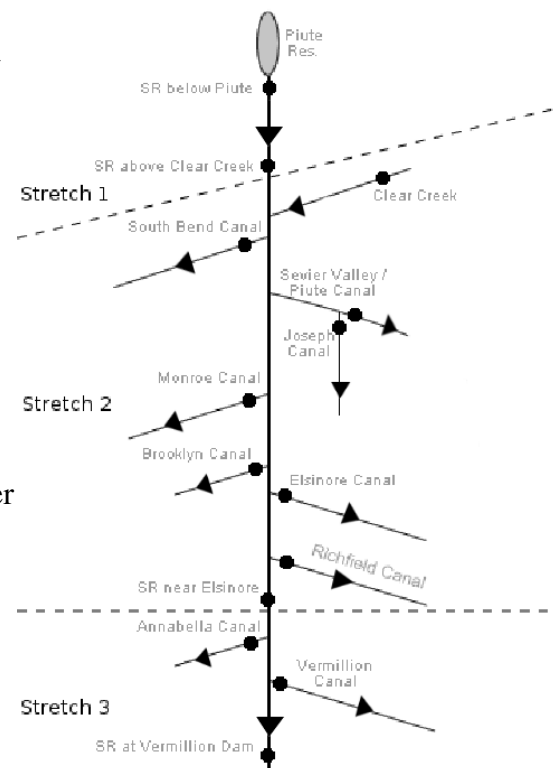


Figure 2. Piute Reservoir to Vermillion Dam Diagram

independently, and the resulting models can be combined serially to obtain a model for the entire system from Piute Reservoir to Sevier River at Vermillion. Hourly historical data from each of the data collection stations on the river begin in January 2000 and continue to the present day. Equipment malfunctions and communication errors have caused missing data points to occur; however, over 98% of the required data are present. Since nearly five years of data are available, we used three years of data to fit the models and two years of data to validate the model.

C. Data Dissemination

To provide water users and water management officials with accurate, timely hydrologic information the SRWUA maintains the webpage <http://www.sevierriver.org>. This website contains detailed information on flows, reservoir heights, weather conditions, and more for the entire Sevier River Basin. As part of his previous work with the Bureau of Reclamation, the author helped design a web-based data display system entitled OpenBasin [10]. This system is the basis behind SRWUA data dissemination process. Website pages of particular interest for this paper include the Upper region page (<http://www.sevierriver.org/upper>), and the Central region page (<http://www.sevierriver.org/central>). The Upper region page displays data for the Piute Reservoir and the Central region page displays data for all of the downstream area that will be used in this system identification procedure. Additionally, real-time model estimates and comparisons for each of the models introduced in the next section can be found at <http://www.sevierriver.org/flow/models.php>.

D. 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. Also, 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 Remote Terminal Units (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 3.

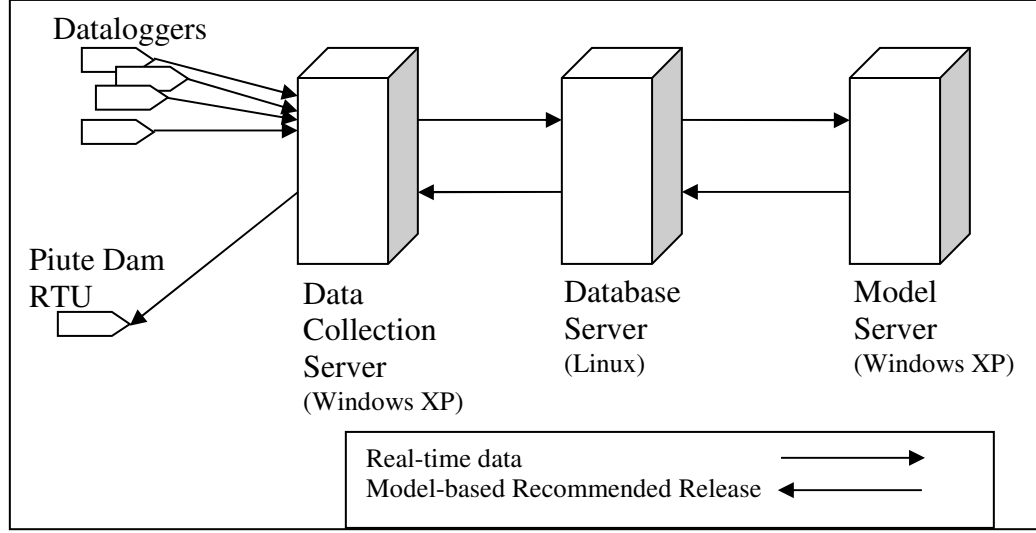


Figure 3. Flowchart of Technology Interconnections

III. MODELING PROCEDURES

This section will focus on the second stretch of the river from Sevier River above Clear Creek to Sevier River at Elsinore (see Figure 2). This stretch of the river poses the most difficulties because it has regulated inflow, seven regulated outflows, unregulated inflow, and significant lag.

A. System Representation

The inflows and outflows of the river stretch are labeled u_1 through u_9 where u_1 is the flow at Sevier River above Clear Creek, u_2 is the flow at Clear Creek, and so forth downstream. There will be no forced sign changes for the different flows. Negative numbers will be treated as outflows while positive numbers will be treated as inflows. The arrows on Figure 2 indicate the direction of flow for each section of the river. An additional step in our validation process is ensuring that the correct signs correlate with correct inflows and outflows.

It should also be noted that Sevier Valley/Piute Canal is u_3 while Joseph Canal is u_4 and the final diversion, Richfield Canal, is denoted u_9 . We let y denote the flow downstream at the point to be modeled. For the second stretch of the river, this location is Sevier River near Elsinore.

Both of the inflows into this stretch, $\{u_1, u_2\}$, are represented as inputs to the model. The outflows of the river, $\{u_3, \dots, u_9\}$, are also modeled as inputs because each diversion along the river from Piute Reservoir to Sevier River at Vermillion Dam has an automated gate. These gates are adjusted to maintain a specified reference flow by independent PID controllers.

Since we can specify the flow at each one of these locations and ensure that the resulting flow will be within a nominal amount of the desired flow there is no need to predict the amount of flow at each of these structures. Consequently, the resulting model type for Stretch 2 is a MISO model.

B. Lag Determination

We determined approximate time lags for each of the inflows/outflows obtained by shifting each input data set u_i by n hours where $n = 0, 1, \dots, 9, 10$ and performing a statistical correlation test on the shifted data versus the downstream flow at Sevier River at Elsinore. The amount n of the shifted data which had the highest correlation was then taken to be the appropriate time lag of that input. The maximum of n was set at 10 for this stretch of the river because it is known that the lag from any u to the downstream y is less than 10 hours. The resulting lags are $L = [3, 2, 2, 2, 2, 1, 0, 0, 0]^T$, where l_i is the appropriate lag for input u_i .

It is important to note the monotonic order of the lags in L is actually representative of the physical layout of the system. The flow u_1 is furthest from Sevier River at Elsinore and u_9 is the closest.

The SRWUA's previous estimate of the lags, based on geographic factors, average flows, and prior experience, was $[5, 4, 3, 3, 3, 2, 1, 1, 1]^T$; however, a comparison of these two sets of lags using a generic mass balance model on the validation data showed that the lags we determined based on statistical correlation were consistently closer to the true downstream flow than those based on SRWUA's approximate lags. After this process of determining the approximate system lags L , we used this information to formulate models representative of the river basin.

C. Model Classes

We compared three different model classes in the process of developing a model for the Sevier River: a mass balance model, a parameterized second-order model with delay, and a parameterized mass balance model. We show the form of each model along with the results of identification and validation below. We used standard linear regression techniques to fit the parameters of the models.

1) Mass Balance Model: A mass balance model simply states that the amount of water flowing into a river will be the same amount that flows out of the river. This MISO mass balance model was adapted from a similar SISO model developed by Weyer in [6].

$$y(k) = \sum_{i=1}^9 a_i u_i(k - l_i)$$

Weyer used a parameterized mass balance model to estimate flows from the height of the water running over a gate. Since the measurement devices for the Sevier River already

calculate flow, we modified the model by fixing the parameters for each flow in the following form:

$$A = [1, 1, -1, -1, -1, -1, -1, -1, -1]^T.$$

The values of 1 as weights indicate inflows; whereas, the values of -1 indicate outflows.

It should be noted that although a mass balance model gives a good approximation of river flows it also has its weaknesses. Due to evaporation, seepage, return flows, unregulated inflows, and other disturbances, the sum of the water flowing from the inflows does not strictly equal the sum of the water flowing through the outflows. Consequently, none of these phenomenons will be captured with a strictly mass balance approach.

2) *Parameterized Second-Order Model with Delay*: The Saint-Venant equations are two partial differential equations that are commonly used to model open-channel dynamics. As shown by Litrico, under simplifying assumptions such as uniform width and depth these equations can be simplified to the diffuse wave equation. Linearizing around a reference discharge results in the Hayami equation. This equation can be sufficiently approximated by a second-order system plus delay [4].

Additionally, in [7], Weyer, et al. show that this model can be fitted using black box system identification methods as opposed to physics-based derivations with comparable results.

Based upon these two results, we chose a second-order parameterized model with delay to approximate the Hayami equation and model the Sevier River Basin.

$$y(k) = b_1 y(k-1) + b_2 y(k-2) + \sum_{i=1}^9 a_i u_i(k-l_i)$$

The parameters obtained using a linear regression on the first three years of data are

$$\begin{bmatrix} b_1 \\ b_2 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{bmatrix} = \begin{bmatrix} 1.2841 \\ -.3004 \\ .0115 \\ .0194 \\ -.0027 \\ -.0144 \\ -.0173 \\ -.0075 \\ -.0034 \\ -.0074 \\ -.0088 \end{bmatrix}.$$

These parameter fittings are reasonable in relation to proper identification of inflows and outflows. u_1 and u_2 are the only inflows into the river system, and a_1 and a_2 are the only parameters greater than zero. This is what we would expect of a properly fitted model.

Additionally, the average weight of the inflow parameters $\{a_1, a_2\}$ at .015 is similar to the average weight of the outflow parameters $\{a_1, \dots, a_9\}$ at .018, suggesting consistency with the mass balance model. However, upon closer analysis, the Sevier River Basin model may not fit a mass balance framework due to unmeasured inflows.

To see this effect, we calculate the average difference between inflows and outflows Q_{diff} as follows,

$$Q_{diff} = \frac{\sum_{k=1}^n \sum_{i=1}^2 u_i(k - l_i) - \sum_{k=1}^n \sum_{i=3}^9 u_i(k - l_i)}{n},$$

where n is the number of data points in the historical data set. For this stretch of the river we determined that there is a Q_{diff} of $-0.37 \text{ m}^3/\text{s}$. This implies that water is consistently gained from unmeasured inflows between Sevier River at Clear Creek and Sevier River

at Elsinore. With an average flow of $6.16 \text{ m}^3/\text{s}$ this represents about 6% of the upstream flow. The Sevier River system clearly does not fit well into a strict mass balance model.

D. Parameterized Mass Balance Model

Using information concerning the Sevier River Basin and techniques from each of the two previous models, we were able to develop a more accurate model for this system.

Along the Sevier River, every device that measures diverted water from the river is located very close to the actual diversion gate. Consequently, we can assume that the measurement for every outflow is exact (i.e. the weight is 1). The only circumstance that would invalidate this assumption would be if there was some problem with the sensor; however, even in this instance we do not want to include those effects in our model since these problems are only temporary.

With the outflow weights, a_3, \dots, a_7 , fixed at unity, we are still able to parameterize the input weights a_1 and a_2 . This is justifiable because the locations that measure the inflows to the river are physically separated from the final measurements downstream. Thus, the accuracy of the inflow measurements is disrupted by evaporation, seepage, and other unmeasured disturbances.

The result of this analysis is a partially parameterized mass balance equation.

$$y(k) = a_1 u_1(k - l_1) + a_2 u_2(k - l_2) + \sum_{i=3}^7 u_i(k - l_i)$$

Using the three years of training data to fit the model we get $a_1 = .9841$ and $a_2 = 1.4164$.

Before continuing we need to validate that these parameters are reasonable. Earlier we discussed that the Sevier River actually gains water as it flows downstream. For this reason it makes sense that the average weight for these two inflows would be larger than

one. Additionally, the weight for Clear Creek is greater than one and the weight for Sevier River above Clear Creek is less than one. This is also to be expected.

Clear Creek is an unregulated inflow. This means that Clear Creek is likely very representative of the conditions around the river. For example, when the snow pack begins to melt off of the mountains in the spring, both the flow down Clear Creek and unmeasured flows will be higher than normal. However, since the Sevier River above Clear Creek is a regulated flow we would expect it to contain very little information about unmeasured inflows. Consequently, it is not unreasonable that the weight of Clear Creek be above one (to include unmeasured inflows) and the weight of Sevier River to be slightly below one (to compensate for evaporation and seepage).

IV. MODELING RESULTS AND ANALYSIS

A. Mass Balance Model

The results of the mass balance model validation are shown in Figure 4. It should be noted that the large errors

between about 4500 hours and 7500 hours are due to sensor error in the validation set and should not be considered representative of this model or any of the other models.

Consequently, the calculation

of error metrics for these models does

not include the hours from 4500 to 7500.

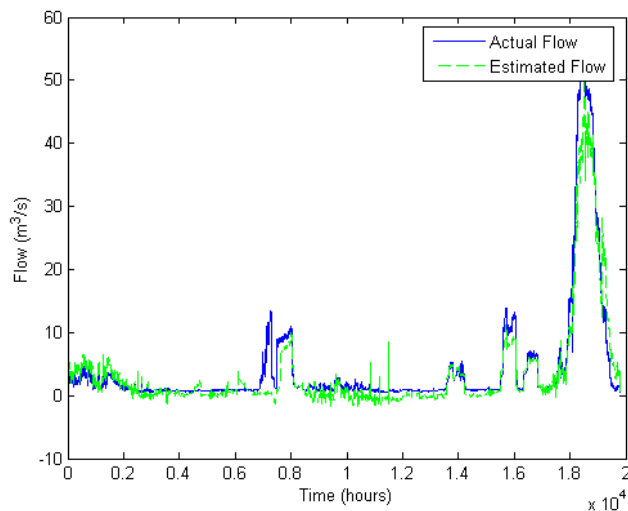


Figure 4. Validation of Mass Balance Model

The mass balance model does a fairly good job of capturing major system events like the large flows at 8000, 14000, and 16000 hours. However, the mass balance model is heavily influenced by erroneous data such as the large spikes near 11000 hours and is not very good at predicting during times of low flow such as the span from 9000 to 13000 hours.

B. Parameterized Second-Order Model with Delay

The validation results of the parameterized second-order model with delay are

shown in Figure 5. This model performs well at capturing general trends of the river system; however, it is not very good at estimating the true flow of the system during periods of higher flow. This model is very good at estimating the flow of the system

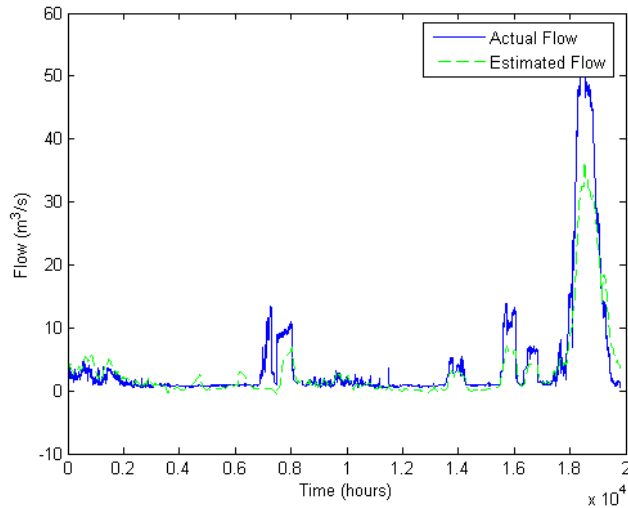


Figure 5. Validation of Second-Order Model with Delay

during times of low flow (9000 to 15000) and especially when the flow is rather erratic (hours 1 to 3000 and 9000 to 11000). This model is also not influenced much by abnormal behavior of the system. This is seen by the model's relatively calm estimates near the time of the large peaks near hour 11000.

C. Parameterized Mass Balance Model

The parameterized mass balance model, as shown in Figure 6, is heavily influenced by erroneous data such as the large spikes near 11000 hours. Nevertheless, it

does a good job of capturing the trends of the river during high flow conditions such as those at 8000, 14000, 16000, and even the flood conditions as 18000. As expected, this model behaves much like the mass balance model; however,

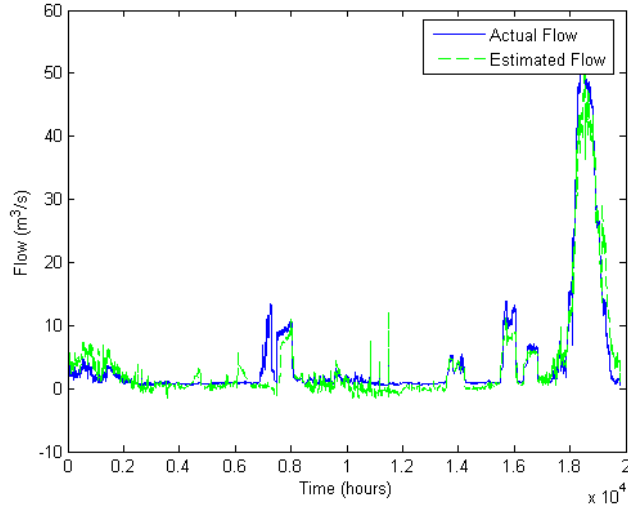


Figure 6. Validation of Parameterized Mass Balance

it usually estimates a little higher than the mass balance model does—especially when there is significant flow in Clear Creek.

D. Comparative Analysis

To accurately compare the system identification methods used we split the validation set into different ranges that are respective of various system conditions. These ranges represent the system in periods of low flow, high flow, rapidly changing flow, and overall flow. Additionally, the final condition of the river is represented in the last two and a half months of data. This condition is that of severe flood conditions during the spring of 2005. The periods of time that are included for each range is summarized in Table 1. The root mean square error and the mean absolute error for each model and range is shown in Table 2.

As can be seen from Table 2, the mass balance model and the parameterized mass balance model perform comparably during most

Period	Range (hours)
Low Flow	1-3000, 9000-13000
High Flow	7700-8200, 15490-16870
Changing Flow	1-2000, 9000-11000
Overall Flow	1-4000, 8000-18000
Flood	18000-19000

Table 1. Ranges of System Conditions

Flow Type	Mass Balance		Second Order		Param. Balance	
Low	1.17	1.05	0.95	0.74	1.30	1.09
High	1.74	1.38	3.57	2.97	1.74	1.35
Changing	1.31	1.18	1.14	0.87	1.50	1.26
Overall	1.13	0.91	1.51	0.94	1.19	0.94
Flood	7.13	6.12	11.2	9.19	5.90	4.95

Table 2. Model Validation Errors: Root Mean Squared(m^3/s), Mean Absolute Error (m^3/s)

conditions on the river. The parameterized mass balance model usually has a little higher error than the usual mass balance model; however, it appears that the parameterized version is slightly better at times of high flow and much better during flood-like conditions. Thus it appears that the parameterized model is best suited for high flow conditions.

The parameterized second-order model with delay performed significantly better during times of low flow and rapidly changing flow than the other two models did. Unfortunately, this model was not very accurate and estimating during times of high flow, and it was especially bad during flood conditions.

From a water management perspective, accurately modeling the river system is most crucial during times of high flow such as the irrigation season. This is because at these times the demand for water and the potential for water conservation are the highest. Thus the most profitable model must estimate high flow conditions well. Consequently, we chose parametric mass balance model as the preferred model for the Sevier River. This model performs at least as well as the other models on high flow conditions and even better than the other two during extreme high flow conditions.

V. CONTROL DESIGN

A. System Model

After completing the system identification process explained above for each stretch of the river, we combine the models serially (e.g. the outflow of stretch 1 becomes the input to stretch 2, etc.) to generate a model for the entire system. With a complete model of the system, we calculate the transfer function from each inflow to the downstream input. These transfer functions are listed in Appendix A. It is important to note that since our model is discrete, the transfer functions are in the z domain. The subsection on controller design will further discuss the importance of this fact.

Once again we validate this model on two years worth of data and get the results shown in Figure 7. From this validation test we can see that the final model accurately describes major system

events. Additionally, since there are no dynamics in the model, the roots of all the transfer functions (Appendix A) are zero. For discrete time systems, stability is defined as $|\lambda_i| < 1$ where λ_i are the roots of the system.

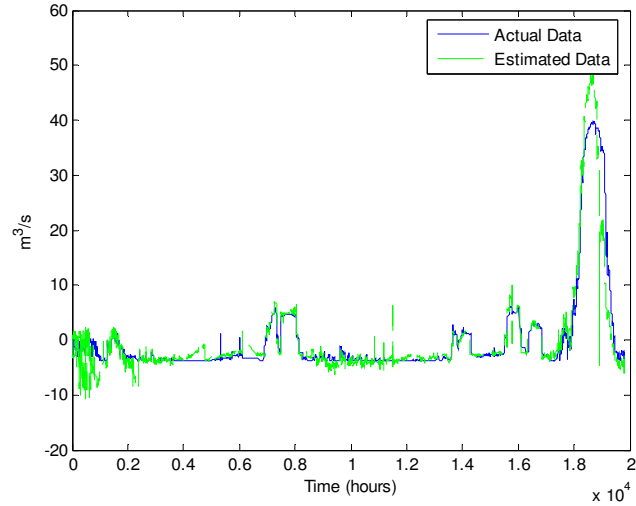


Figure 7. Validation of System Model

This condition is clearly satisfied for each root; consequently, the model is stable. With this confirmation we accept the current model and proceed with controller design.

B. Controller Design

1) *Feedback Explanation:* Feedback control is a mathematical mechanism which alters the input of a

system based upon the difference between the actual system output and

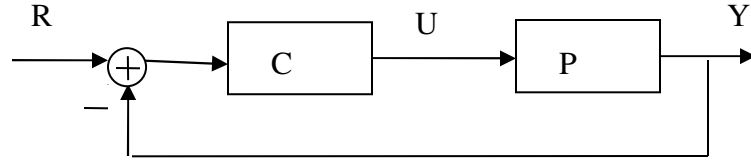


Figure 8. Sample Feedback System

the desired output, or reference command. A sample feedback system is illustrated in Figure 8. The R denotes the reference command, or the desired output of the system. The P denotes the plant, or system that we are trying to control and Y is the actual output of the system. Notice that the difference between the current output, Y , and the reference command, R , is passed into the controller, C . The controller must decide what input, U , to give the system based upon this difference.

The main purpose for a feedback controller on the Sevier River is to correct for unknown disturbances in the system. For example, during certain periods of the year, snowmelt runs into the river through many unmeasured creeks. By detecting the presence of this surplus water, the controller could reduce the water released from the reservoir while still maintaining downstream flow levels. In this manner, a feedback controller can compensate for either a disturbance including too much water, too little water, and modeling errors.

2) *Controller Requirements:* The primary specification for the controller is that it asymptotically tracks a reference signal (i.e. rejects disturbances until they are zero). Asymptotic tracking requires a free integrator in the controller. Additionally, since the controller will only receive measurements and execute commands every hour, both the

controller and model must be discrete. Additionally, since the reservoir release at Piute Dam is the controllable inflow, it will be the only input for which we design a controller.

3) *Controller Design:* We begin the controller design process by defining our controller to be $k_i/(z-1)$ where $1/(z-1)$ is the definition of a discrete integrator, and k_i is an adjustable gain for the integrator. This simple controller alone will provide asymptotic tracking; however, it is wise to choose k_i suitably to get improved system responses. We find that $k_i = .0275$ creates a stable feedback system with a fairly accurate response time and little overshoot or oscillations as shown in Figure 9.

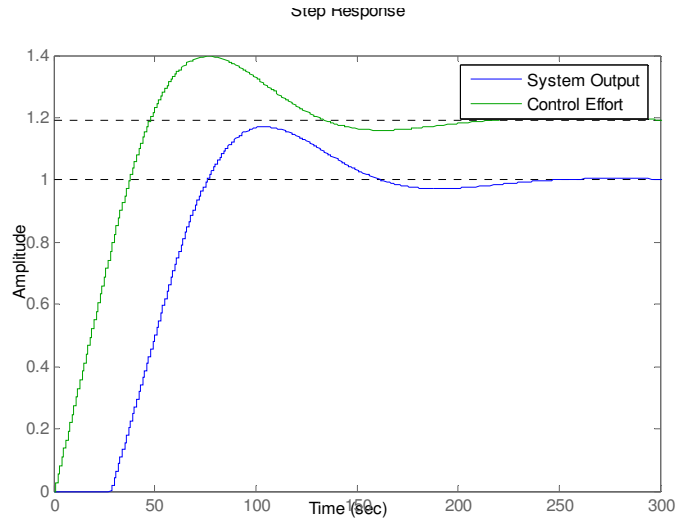


Figure 9. Step Response of Feedback System for $k_i = .0275$

Notice that the control effort (green line) initially starts at zero and then gradually increases. This is how an integrator works; however, for a reservoir modeling application it is more beneficial to begin with a control output roughly equal to the desired downstream amount and then control asymptotically from there.

This can be accomplished by observing that the gain K_i for each transfer function in our system is simply the sum of the numerator's coefficients. Consequently, the quantity $1/K_i$ would be the appropriate gain to add to an input system to achieve perfect output in the absence of modeling error and disturbances. For example, the gain K_i for the transfer function between Sevier River below Piute Reservoir and Sevier River at Vermillion Dam is $(.134 + .4025 + .3018) = .8383$. Thus we would want to feed

the reference input R forward with a gain of $(1 / .8383) = 1.1929$ directly into the model. This allows the controller to take immediate action in response to a change in the reference command. The step response for this

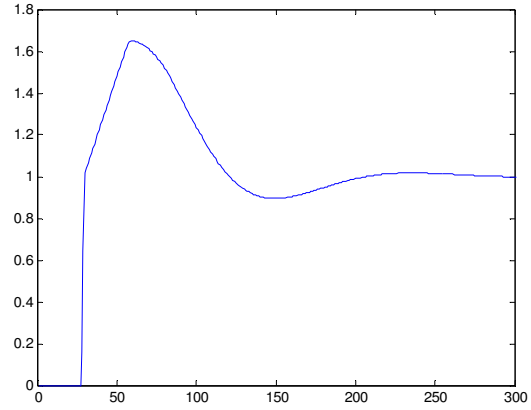


Figure 10. Step Response of Feed-forward System

10. Notice how even after reaching the target system output, the controller continues to increase the input until the system overshoots by 60%. That is unacceptable performance.

The error with this controller is that the integrator continues increase during the first 29 time steps of lag. Consequently, by the time the system responds to the initial step function, the integrator already has a large error associated with it. This causes the control force to continue to rise and create a massive overshoot.

The solution to this problem is to reset the integrator each time the reference signal changes. More precisely, the integrator must be reset to zero n timestamps after the reference is changed, where n is the amount of lag in the system. This formulation is equivalent to placing a delay of n between the reference and the feedback junction as shown in Figure 11.

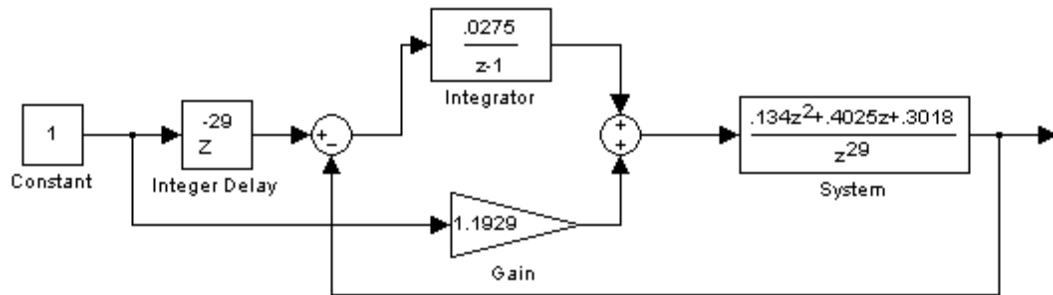


Figure 11. Block Diagram of Delayed-Integral Controller

Following this control design produces a step response without overshoot as shown in Figure 12. This controller will be the one used throughout the rest of this document.

C. Controller Simulation

After designing a controller according to specifications, we simulate the controller/model combination to see how closely they match the real system. This is done by using the validation data that we tested the model upon previously; however, this time we use the downstream flow at Vermillion Dam as the reference command. With this configuration, the model and the controller will interact in a manner that attempts to recreate the two years of validation data. We can judge the model and controller's performance based upon how well correlated the results from the controller and actual data are. The results of this verification test are shown in Figure 13. The green line is the action performed by our controller, and the red line is the actual reservoir releases for this period. The blue line is the amount of water that our model predicted will be flowing downstream at a given time, and the black line is the amount of water that was actually flowing at that time. Notice how close the red and green lines are to one another, and similarly how close the blue and black lines are. This correlation indicates that the model used is sufficiently expressive of the real dynamics of the system and that the controller is able to interact with the model to provide performance comparable to the validation data.

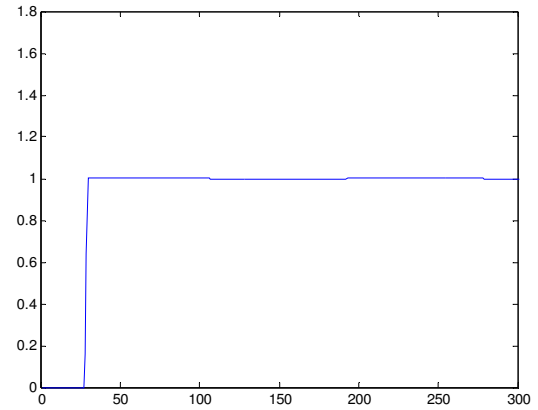


Figure 12. Step Response of Final Controller

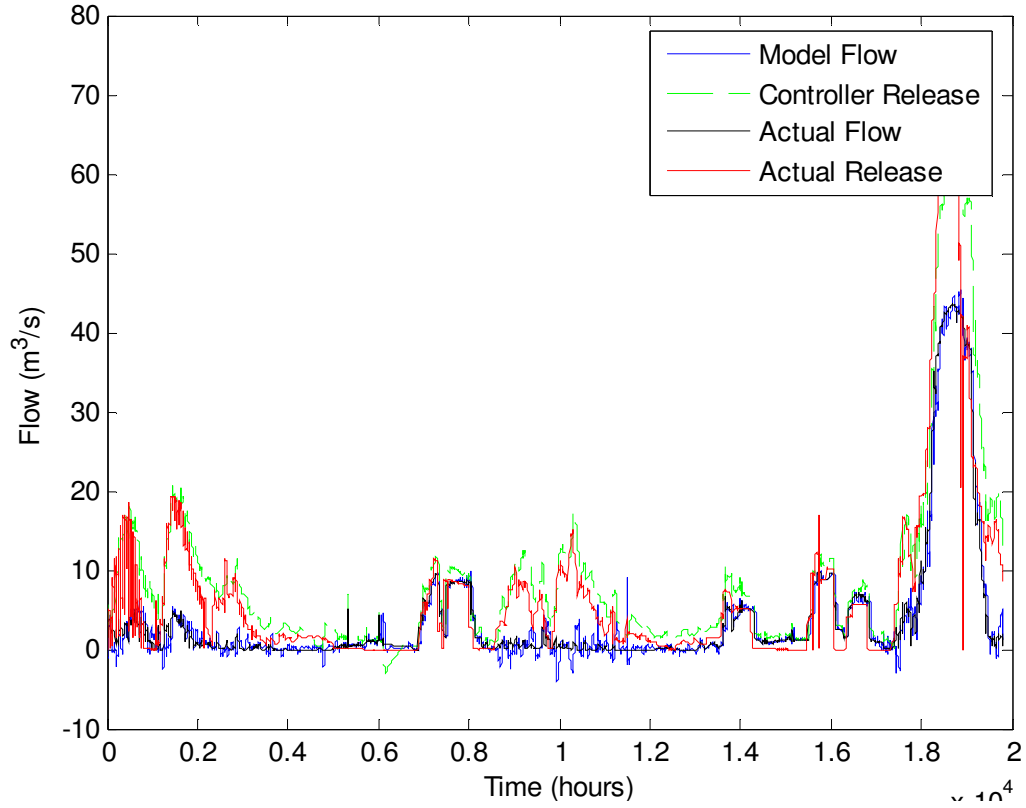


Figure 13. Validation of Controller

VI. IMPLEMENTATION

A detailed description of the implementation of automation technology for the Piute Dam is contained in [11]. The following section will highlight relevant details of that paper.

A. 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.

1) 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.

2) 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).

3) 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.

B. 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.

1) 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.

2) *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. A screenshot of the online supervisory controls is show in Figure 14.

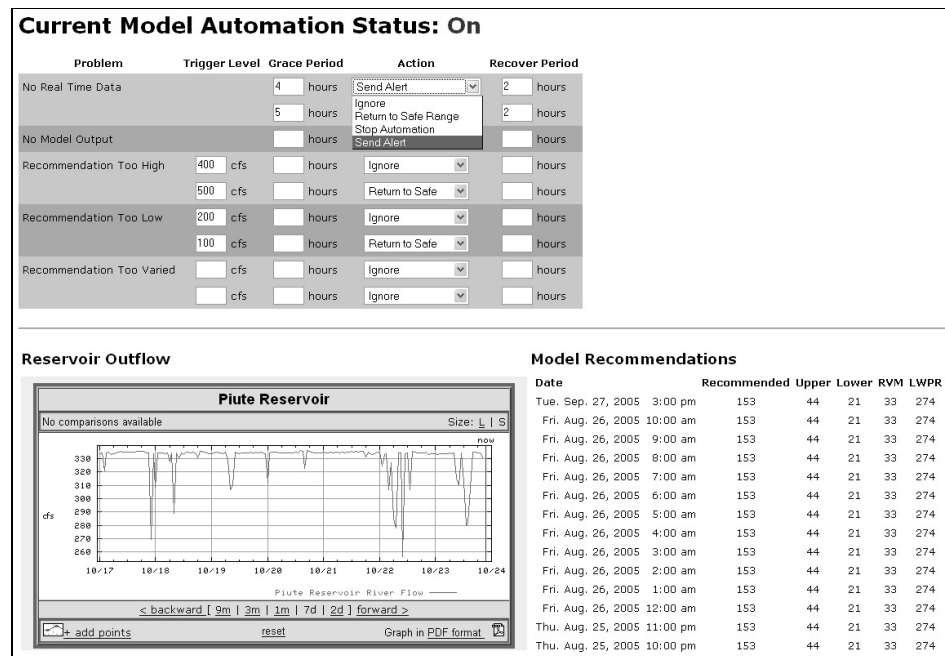


Figure 14. Screenshot of Online Supervisory Control Features

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.

3) *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.

C. 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.

1) 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.

2) *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 proactively 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.

3) *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.

4) *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.

VII. CONCLUSION

In this thesis, we have provided a thorough description of the Sevier River system from Piute Reservoir to Vermillion Dam. We have developed a modeling procedure to identify this river system and have found it favorable in comparison to other popular results. We constructed a model for the entire system and built a discrete controller according to specifications including asymptotic tracking. Finally, we have described the automation framework that has been built to facilitate the real-time model-based automation of Piute Dam. Also, we have included possible expansions and improvements for both the automation software and for the controller.

The entire system described in this thesis including the OpenBasin software, the system controller, and the automation implementation will be tested on the Piute River in the spring of 2006. As discussed throughout the thesis, there are many improvements which could be made to the system to make it more robust and convenient. Nevertheless, the central algorithms are in place to allow an informative and worthwhile performance trial to be conducted in the spring.

It is my desire that this automation system, and future developments derived from it will provide an efficient and convenient mechanism for better water management and conservation.

REFERENCES

- [1] P.O. Malaterre and B.P. Baume, "Modeling and regulation of irrigation canals: existing applications and ongoing researches", in Proceedings of IEEE Conference on System, Man, and Cybernetics", San Diego, CA, USA, 1998, pp. 3850-3855.
- [2] X. Litrico, D. Georges, and J.-L. Trouvat, "Modelling and robust control of a dam-river system", in IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA, 1998, vol. 4, pp. 3862-3867.
- [3] X. Litrico, V. Fromion, F.-P. Baume, and M. Rijo, "Modelling and PI control of an irrigation channel", in Proceedings of European Control Conference ECC03, Cambridge, UK, 2003.
- [4] X. Litrico, "Robust IMC Flow Control of SIMO Dam-River Open-Channel Systems", in IEEE Transactions on Control Systems Technology, Vol. 10, no. 3, pp.432-437.
- [5] X. Litrico, and V. Fromion, "Advanced control politics and optimal performance for an irrigation canal", in Proceedings of European Control Conference ECC03, Cambridge, UK, 2003.
- [6] E. Weyer, "System identification of an open water channel", in Control Engineering Practice, vol. 9, pp. 1289-1299, 2002.
- [7] S.K. Ooi, M.P.M. Krutzen, and E. Weyer, "On physical and data driven modelling of irrigation channels", in Control Engineering Practice, vol. 13, issue 4, April 2005, pp. 461-471.
- [8] E. Weyer, "LQ control of an irrigation channel", in Proceedings of the 42nd IEEE CDC, Hawaii, USA, pp. 750-755, 2003.
- [9] Y. Li, M. Cantoni, and E. Weyer, "Design of a Centralized Controller for an Irrigation Channel Using H_∞ Loop-shaping", in Control 2004, University of Bath, U.K., September 2004.
- [10] B. Berger and M. Maxwell, "OpenBasin: An Open Source Software Framework for Real-timeWeb-based Hydrologic Data Dissemination", in Proceedings U.S. Committee on Irrigation and Drainage Conference on Water Management, Salt Lake City, UT, USA, Oct. 2004, pp. 365-373.
- [11] M. Maxwell, "Real-Time Model-Based Dam Automation: A Case Study of the Piute Dam", in Proceedings U.S. Committee on Irrigation and Drainage Conference on SCADA Technologies, Vancouver, WA, Oct. 2005, pp. 49-58.

APPENDIX A

TRANSFER FUNCTIONS

Transfer function from srps to srv:

$$\frac{.134z^2 + .4025z + .3018}{z^{12}}$$

Transfer function from ccd to srv:

$$\frac{.5049z + .7844}{z^{12}}$$

Transfer function from msbc to srv:

$$\frac{-.3565z - .5538}{z^{12}}$$

Transfer function from svpc to srv:

$$\frac{-.3565z - .5538}{z^{12}}$$

Transfer function from jch to srv:

$$\frac{-.3565z - .5538}{z^{12}}$$

Transfer function from mch to srv:

$$\frac{-.3565z - .5538}{z^{11}}$$

Transfer function from bch to srv:

$$\frac{-.3565z - .5538}{z^{10}}$$

Transfer function from ech to srv:

$$\frac{-.3565z - .5538}{z^{10}}$$

Transfer function from rch to srv:

$$\frac{-.3565z - .5538}{z^{10}}$$

Transfer function from ach to srv:

$$\frac{-1.692}{z^8}$$

Transfer function from vch to srv:

$$-1$$

srps:	Sevier River below Piute Dam
ccd:	Clear Creek
msbc:	Monroe/South Bend Canal
svpc:	Sevier Valley/Piute Canal
jch:	Joseph Canal
mch:	Monroe Canal
bch:	Brooklyn Canal
ech:	Elsinore Canal
rch:	Richfield Canal
ach:	Anabella Canal
vch:	Vermillion Canal
srv:	Sevier River at Vermillion Dam
