

AUTOMATING PIUTE DAM AND THE EIGHT CANALS THAT IT SERVES

Blake E. Durtschi¹

ABSTRACT

Piute Dam and Reservoir are located on the upper Sevier River in south central Utah. The reservoir has a capacity of 60,000 AF and serves eight canal companies located around the city of Richfield. Because of lag times between the reservoir and the various canals and an unregulated inflow below the dam (Clear Creek), setting reservoir releases is an art (and not always successful). Too much or too little water at the lowest canal diversion is a problem.

Several years ago, a gate actuator was installed on Piute Dam outlet works. Telemetry to the dam was also established. This summer, the Piute Reservoir and Canal Company and Bureau of Reclamation will be fully automating the dam. The focus of this work has been to design a forecasting model for use in calculating a suggested reservoir release. This decision-support tool uses hourly measurements from a real-time monitoring system (www.sevierriver.org) to design a control algorithm for the reservoir release. The most efficient controller is then determined through simulation. This paper shows how the model and controller were implemented using the existing SCADA (supervisory control and data acquisition) system for the 2009 irrigation season. The benefits from using the controller will be estimated.

Demand forecasts for the individual canals will be provided using the existing website of the Sevier River Water Users Association (www.sevierriver.org). This Internet-based system will also be used to control the individual gates on the eight canals. Proper security is being installed to protect the integrity of the automation system.

INTRODUCTION

The Sevier River, in central Utah, serves primarily as irrigation water with a small amount used for municipal water in the local towns. The Sevier River Basin is completely enclosed so that any water flowing down the river empties into the desert. The Piute Dam and reservoir are located on the upper portion of this river. Water released from the reservoir may be diverted into one of several canals. Any water not diverted into the canals continues down the river and is lost for any other use. This makes the Sevier River a great location to practice water conservation using automated control. Water can be conserved by designing a control algorithm which releases just enough water from the reservoir that every canal gets the water it needs, but no more. Determining how much water to release is made difficult because of large delays in the time it takes the water to move down the river, uncontrolled inflows, and various environmental factors which affect the amount of water such as evaporation, seepage, or rainfall.

¹ Graduate Student, Brigham Young University, and Computer Assistant, Bureau of Reclamation, Provo UT

Figure 1 shows the stretch of river below the Piute reservoir. The release from the dam determines how much water enters the system. There is also an uncontrolled, but measured inflow at Clear Creek, and eight diversion canals that take water out of the river. At the end of the river is the small Vermillion Dam. The goal is to have no water flowing past Vermillion Dam as it will be lost to our water users.

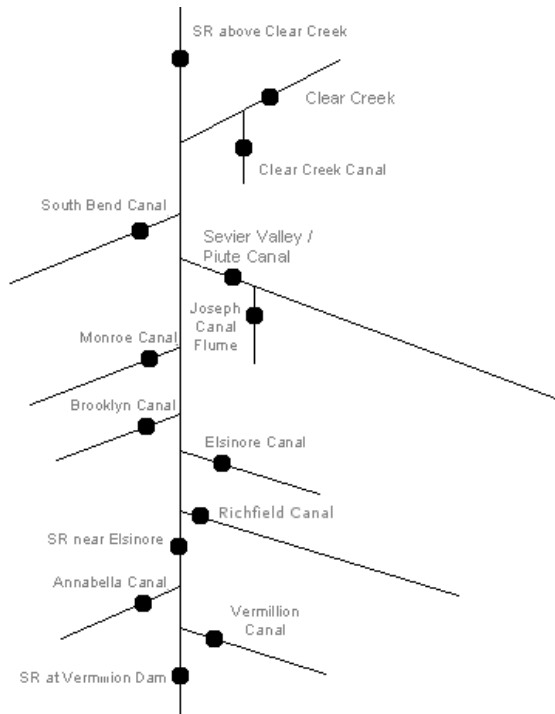


Figure 1. Stick figure showing the stretch of the Sevier River below the Piute Dam.

Table 1. Delays associated with each station along the Sevier River.

Station Name	Symbol	Delay (hours to bottom of river)
Piute Reservoir	srps	32
Sevier River above Clear Creek	srcc	18
Clear Creek	ccd	18
South Bend Canal	sbch	16
Sevier Valley Piute Canal	svpc	15
Joseph Canal	jch	15
Monroe Canal	mch	14
Brooklyn Canal	bch	13
Elsinore Canal	ech	12
Richfield Canal	rch	12
Sevier River near Elsinore	sre	12
Anabella Canal	ach	8
Vermillion Canal	vch	0
Sevier River at Vermillion	srv	0

Each black circle in figure 1 shows where there is a station measuring flow. Each of the stations on an out-flowing canal also have a locally controlled gate which regulates the flow to match a flow value set by the canal owner. There are also four measuring stations along the river, one at the reservoir release (not shown), another just above Clear Creek, one near Elsinore, and one measuring flow past Vermillion Dam. Flow data is collected at all of these stations every hour and stored into a central database. This remote monitoring and telemetry has been installed in the system since the summer of 2000.

Much of the previous work in canal/river automation has been focused on obtaining accurate measurements of the canal/river and manually controlling gates remotely. Few rivers or canal systems in the world have a computer determining how to control the gates releasing water from various reservoirs, this is groundbreaking work. Another reason few canal/river systems are controlled this way is because individuals can be skeptical about having their livelihood controlled by a computer. There are, however, two groups that have had success in controlling rivers and canals using computers: one in southern France, and one in Australia.

The group in southern France (Litrice, Malaterre, et. al.) has designed various algorithms, including robust control and proportional-integral (PI) algorithms, to control gates on different canal systems [1], [3]. They have also done some work with canal modeling using St. Venant equations for water flow [4]. The group in Australia, led by Erik Weyer, has modeled canals using a mass balance model and designed several different controllers including PI, and linear quadratic Gaussian control (LQG) to control canal gates [6], [7].

None of these approaches work very well with our system. First, the work has been done on canals that have a very small grade and accurate flow measurements. The flow measurements on our river are not very accurate and may be off by as much as 10%. Also our river has steep grade which causes the water to flow more quickly than in the canals which changes the system dynamics. Second, the amount of time the water takes to move down stream (delay) in our stretch of river is over a day and we only receive data once an hour. Previous work has been on canals with much less delay and data collection every second or minute. Because of these differences we will be using a different model than these groups.

The following sections describe the mathematical modeling of the Sevier River and the control algorithm design. Then it is shown how this controller is implemented into the existing SCADA system. Specifically, the system was expanded to include a control algorithm which could automatically set the flows of the reservoir and canals, and to include a system to allow canal owners to input their future water orders.

MODELING THE SEVIER RIVER

Determining the Delay

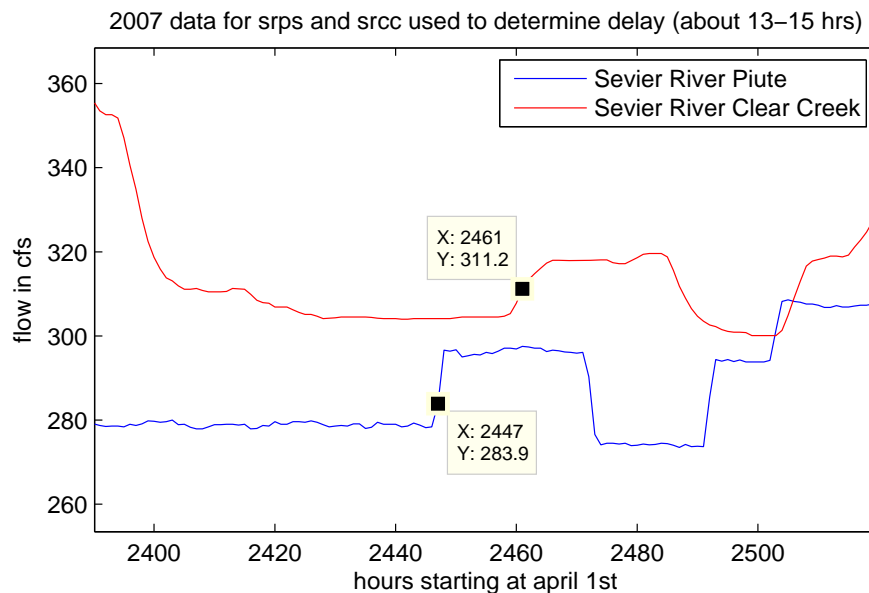


Figure 2. The delay for the stretch of river between the reservoir and Clear Creek is found by finding the corresponding changes and measuring the delay.

To determine the varying amounts of delay in the system we look at flows from two points along the river and look for corresponding changes. We took data from 2007 and found several places in the data where there were large fluctuations in the flow and averaged the delay time to find the different delays in the system. This technique is shown in figure 2. To find the delay of the canals we use the same technique, however we look for places where the upstream river flow stays constant, the canal has a large shift, and the downstream river flow shifts in the opposite direction of the canal flow. Table 1 shows the delays for each part of the system.

Selecting Data

Since water conservation is most important during the irrigation season the model needs to be effective in describing the river during the spring and summer months. River data during the months of April through September for six years (2000-2004, 2006) will be used to train the model, and data from those same months for 2007 and 2008 will be used to validate the model. Data from the year 2005 was not used because that year was a flood year and the flows are unlike any other year. When that year is used for training it causes the model to predict too much water for the other years. It is possible to use 2005 data to learn a model which can be used in flood years, though it has not been done here.

Model Selection

The first step in modeling the Sevier River is to select a model class that captures the dynamics of the river. Previously, Maxwell compared several different models of the Sevier River [5], and concluded that a parameterized mass balance model most correctly describes the river during summer months when river flows are high. We select a parameterized mass balance model as the basis of our model class and add that to a third order system in order to capture some of the smoothing effects of water flow.

We model the Sevier River as a multiple input single output (MISO) system with the flow past Vermillion Dam being the system output. We treat the river inflows and the outflows to the canals as inputs because each represents an external influence on the water in the river. The coefficients to the outflows will be fixed at negative one. This is reasonable since water is being taken out and it is measured as it leaves the system. We let the coefficients to the inputs be variables to be learned from the data, to allow the model to account for evaporation, seepage, or other inflows or outflows that may occur along the river before the water reaches the bottom.

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) + b_1 u_1(k-d_1) + b_2 u_2(k-d_2) - \sum_{i=3}^9 b_i (k-d_i) \quad (1)$$

Our model (1) states that the output at time k , $y(k)$, is determined by the previous three outputs as well as the positive and negative flows added by the inputs. We define inputs in the order that they affect the system, u_1 as the reservoir release, u_2 as the inflow at Clear Creek, and u_3, \dots, u_9 as the other canals. We also define d_i , ($i = 1, \dots, 9$) to be the delay of the river from input i to the end of the river.

To select a model from our specified model class we use an iterative maximum likelihood minimization method to select model parameters for our model. The parameters of our model are:

$$\begin{bmatrix} a1 \\ a2 \\ a3 \\ b1 \\ b2 \end{bmatrix} = \begin{bmatrix} -.7949 \\ -.4595 \\ -.1705 \\ .839 \\ 1.973 \end{bmatrix}. \quad (2)$$

These parameters seem reasonable. Both of the inputs have positive coefficients. Also the larger coefficient from Clear Creek could be explained by it being an unregulated inflow and could represent all other unmeasured inflows along the river which would seem to be higher if the Creek is higher and lower otherwise. The coefficient less than one from the reservoir could be explained by the effects of evaporation and seepage of the released water all down the river.

In order to get a better fit for the model it was necessary to add a filter to the model to smooth out the output. A low pass filter was designed to minimize the mean squared error of the model when compared to the validation set. The filter gives a reduction in root mean squared error from 21.8 to 21.1 yielding the final model of:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (3)$$

where

$$A = \begin{bmatrix} -0.7949 & -0.4595 & -0.1705 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.0952 & 0 & 0 & 0.9048 \end{bmatrix}, \quad B = \begin{bmatrix} 0.839 & 1.973 & -1 & \dots & -1 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

$$C = [0 \quad 0 \quad 0 \quad 1].$$

Model Validation

Figure 3 shows the output predictions of our model for the validation set. The top graph is 2008 data and the bottom graph is 2007 data. Table 2 shows the absolute and root mean squared errors for both years and together.

The model definitely captures many of the trends and changes for both years, but also has periods where its predictions are off especially at the beginning and the end of the season. One limitation on performance is the reliability of the flow measuring devices. The measurements of flow are rated to be within 10% of actual flows. For the flows on this

river that can easily be 5 to 10 cubic feet per second. Over several canals this can add up to a large variance. Also, when flows are at zero or close to zero the sensors may still report flows due to sediment build up in the sensor.

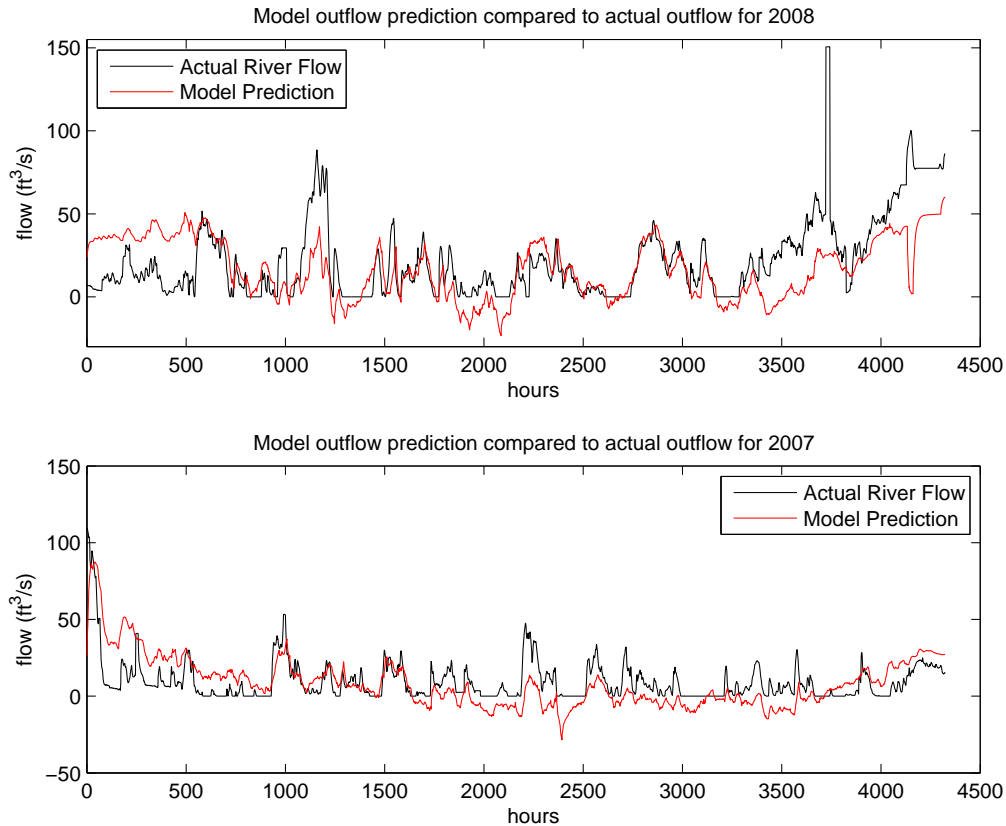


Figure 3. The model output compared with the actual river output. The top graph is 2008 data, while the bottom graph is 2007 data.

Table 2. Error of Model on Two Validation Sets

Year	Root Mean Squared Error	Absolute Error
2008	21.67	15.30
2007	13.37	10.62
2007-2008	17.80	12.96

Another reason for some of the error is that we allow the model to predict negative values. The river cannot have a negative flow, but a negative prediction is the same as predicting a shortage for the last canal. Thus a flow of -10 cfs means that the last canal has 10 less cfs than ordered. Because we did not keep track of how much water was ordered for each canal, it is difficult to determine how much shortage there was in the river when the flow was at zero. So, when the river is at zero and the model predicts some negative value, the negative prediction could be close to the actual state of the system.

There are a few possible explanations for the models poor performance at the beginning and end of the season. At the beginning of the season many of the canals are not being used and have a flow of zero but because of sediment the sensors may be recording one, this can cause unusually large errors. Also, some of the canals turn off before the end of the season which could contribute to large errors at that time as well. Typically more rainfall occurs at the beginning and end of the season, this would cause more water to be entering the system at those times. Another explanation for the error could be temperature which has a great impact on evaporation. During the beginning and end of the season the temperature is considerably lower than the middle of the season.

We feel that even with the error it is still a good model given its relative simplicity. Even when there is some error the model still captures the trends and seems to be off by a constant factor. This leads us to believe that we will be able to control this river and that feedback control can improve system performance

CONTROLLER DESIGN

The objective of the controller for the Sevier River is to select the reservoir release, u_1 , in order to cause the output to match some desired reference signal. On the Sevier River the goal is to minimize water waste so the reference signal for the output of the system will generally be set at some small amount or zero, and stay constant the majority of the time. The controller will then have to decide how much water to let out so that water is delivered to all the canals and the outflow at the end remains low.

Control Architecture

For our controller design we are going to use the two part control architecture shown in figure 4. The first part of the controller is a feed forward controller (F in the figure). It uses any a priori information that we have available in order to determine a close estimate of what the control input should be. For the Sevier River this information includes the reference signal as well as future water orders for the canals. The second part of the controller is the feedback controller (K in the figure). This portion of the controller uses error from the system output in order to ‘fine tune’ the estimates from the feed forward controller.

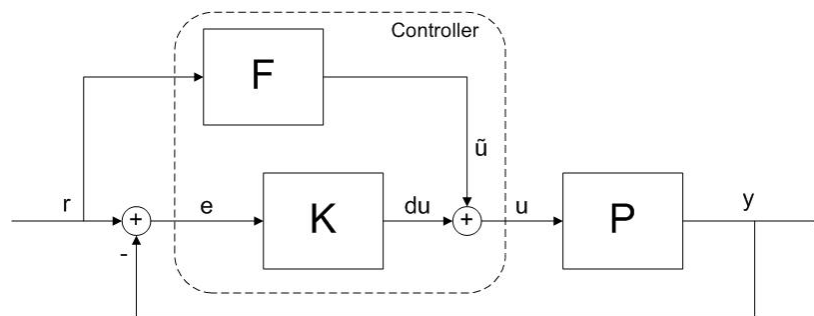


Figure 4. The control algorithm is split up into two pieces. The feed forward controller, F, uses all available information to get an approximate control choice, \tilde{u} . The feedback controller, K, uses the error, e, to adjust the approximation by du.

The control algorithm should be able to control for step changes in the output of the river or in the orders of any of the canals. Because of the large delays in the system it will be difficult to have the algorithm track these changes quickly, but it must eventually control the flow to the right value. The control algorithm should also be robust to disturbances in the flows of the river (such as rain or high evaporation), incorrect measurements, and unexpected changes in the canal flows. The control algorithm must still work if the model parameter choices are incorrect or if the delay in the river is different than we have modeled.

Control Design

First we design the feed forward, F , part of the control algorithm. In order to asymptotically track step changes in the output, the final value of output needs to be equal to the reference command. Using knowledge of the final value theorem², F is selected to be the inverse of the final value of the system, P . Thus $F = P(1)^{-1}$, where $P(1)$ is the discrete time transfer function of the system evaluated at $z = 1$, which is also the final value of the system.

Since the model developed in the previous section represents the river the best we can, that model is used to determine $P(1)^{-1}$. The transfer function for the model is

$$P = \frac{.0952z^2}{(z - .9048)(z + .533)(z^2 + .2619z + .3199)} \quad (4)$$

Applying the final value theorem to the model yields a final value of .412. So F then becomes, $F = 1/.412 = 2.426$.

The feed forward controller also receives the future water orders from the different canals as estimates of the future canal flows. It uses these orders in order to predict what will happen to the canal inputs to the plant. From (3) notice that the inputs are each multiplied by a gain and added to the system. This controller takes each canal order, multiplies it by the inverse of the gain from the model, and subtracts it from the control input. This has the effect of changing the reservoir release early, by the same amount that the input will affect the flow in the river later on.

Next the feedback part of the control algorithm needs to be designed. One of the most common feedback controller designs is the proportional integral (PI) controller which has the property of tracking step changes in the output. The PI controller designed for this system is of the form,

$$K = k_p + \frac{k_i}{z - 1}, \quad (5)$$

² See *Optimal control theory: a course in automatic control theory* by Robert Pallu de La Barrière for an in depth explanation of the final value theorem and its importance in controller design

where k_p is the proportional constant or proportional gain, k_i is the integral constant or the integral gain, and z is the discrete Laplace variable. These two gains must be carefully selected in order to achieve desired control results. If the gains are too high our controller may become unstable, too low and the controller will not achieve desired results.

Following a standard process for selecting these two gains results in $k_p = 1$ and $k_i = .05$. These parameters give the controller a gain margin of 2.69 and a phase margin of 139 degrees. These margins are sufficient for the closed loop system to be stable even with the large delays in our system. Figure 5 shows that the closed loop system with the controller is stable even with a delay of eight hours more or eight hours less than we have estimated. The performance improves with less delay and is worse with more delay.

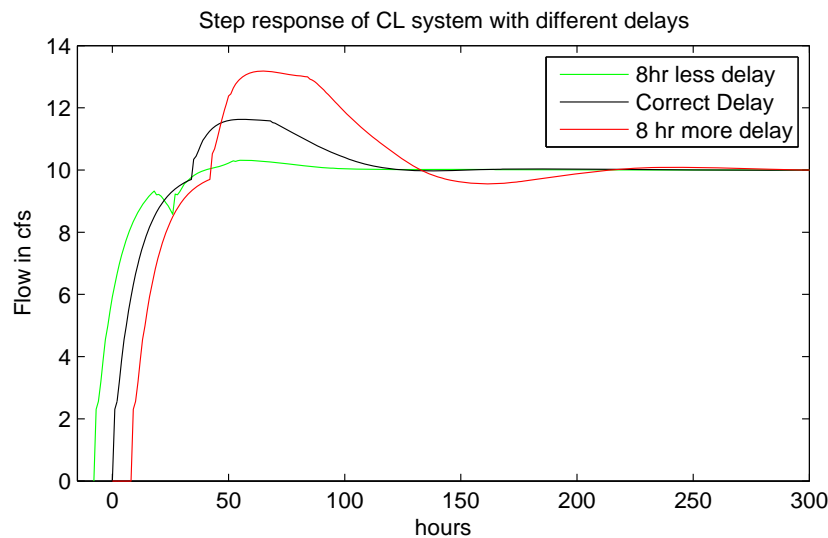


Figure 5. Closed loop system response for the scenario where the actual river delay is 8 hours less than estimated, the same as estimated, or 8 hours more than estimated.

CONTROLLER VALIDATION

In this section the controller is shown to remain stable even with outside disturbances and inaccuracies in the model. This is known as robust stability. The performance of the controller remains acceptable with disturbances and inaccurate model parameters as well.

Disturbance in Canal Input

The first type of disturbance to consider is if a canal takes more or less water than is ordered. This can happen when water users take more or less water than they requested or if the canal measurements are incorrect. It can also happen if the true impact of a canal input on the river flow is not -1 as in the model, but some other value. This would mean that the parameter for the canal input is incorrect. Figure 6 shows the controllers response to this disturbance. At 100 hours a 10 cfs step input to a canal occurs. This canal disturbance has no order attached to it and so there is no way for the feed forward controller to compensate for the disturbance. There is a delay of 20 hours before the disturbance affects the river output. At that point the controller begins to make changes but the river flow continues to drop for 32 hours since it takes that long for the changes to

reach the end of the river. From the time the changes start to affect the river output it takes approximately 35 hours to correct back to within 10% error of the reference.

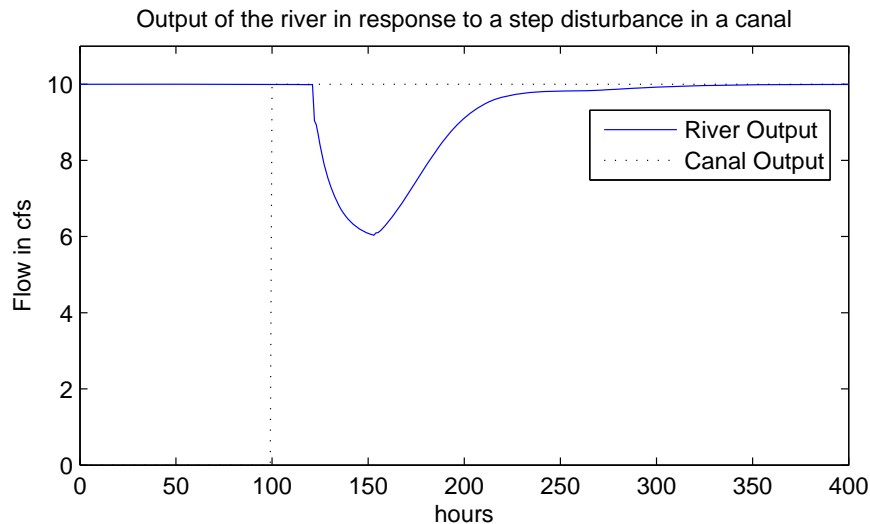


Figure 6. Output of the river with the controller implemented if at 100 hours a canal suddenly takes 10 more cfs without ordering it

The performance of this controller at rejecting disturbances in inputs seems disappointing, taking over 100 hours to correct from a step disturbance. While this is slow, the controller performed well for several reasons. First, stability is maintained in the controller. Second, the disturbance in the canal was 10 cfs, yet the error in the outflow was mitigated by 60%. Third, this is an extreme situation where the disturbance was held out indefinitely. Usually when users make errors in the order it is corrected within a day, which would help the performance of this controller.

Sensitivity to Input and Output Noise

High frequency white noise is not too damaging to the system. In fact we have assumed that many of our measurements are going to have noise in them. Figure 7 shows the output of the system when noise is added to the input and the output. Input noise is added to the reservoir release and is shown in the top graph. Output noise is added to the output measurement and is shown on the bottom graph. The noise is added to a simulation with a step input at time zero and a step disturbance on a canal at time 500. The system remains stable although the controller is unable to correct for the noise. General performance does not seem to be slowed down due to the noise. The rise time and the disturbance rejection do not take any longer.

Simulation on 2008 Data

As a final method of validating the controller it was tested on the true river data for the summer of 2008. The desire was to validate whether the controller remains stable and meets desirable control objectives with noisy data. Another goal was to say what the controller would have done if it had been running in 2008 and how much water would it

have saved. Since there is not any order data for 2008, the actual amount of water that was flowing out of the canal is used as its order. Then that order is given to the feed forward controller with enough advance that it can make any changes it needs. Figure 8 shows the output of the controller with the model when using 1 cfs as the reference signal and when using 5 cfs as the reference signal.

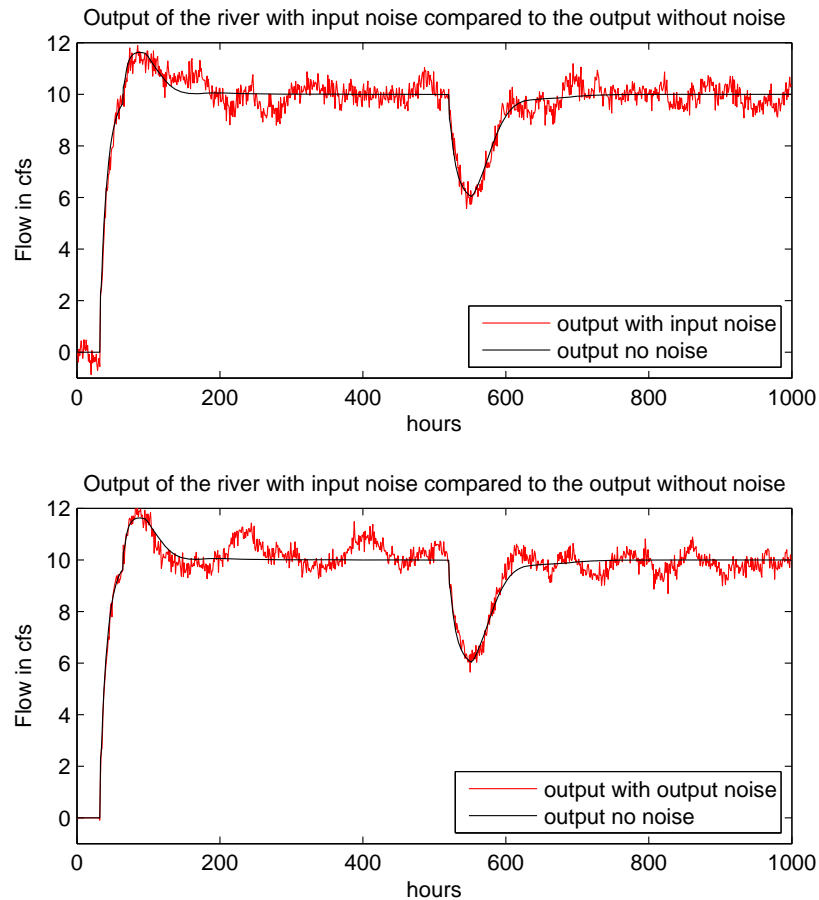


Figure 7. Response of the river to input noise (top) and output noise (bottom). Noise does not affect the overall controller response but does affect the final settling error.

From figure 8 it is easy to see that the controller would have conserved water compared to the actual flow in 2008. The figure also shows the flow of water going negative at times. This represents a shortage to the last canal. This is not very desirable. In fact, it may be more desirable to waste a little water in order to not have a water shortage. One way to do this is to increase the reference command until the controlled output goes below zero infrequently. Table 3 shows the water savings and water shorted by running our controller with varying reference points. As the controller tries to hold the river release closer to zero more water is saved, however more water is shorted as well.

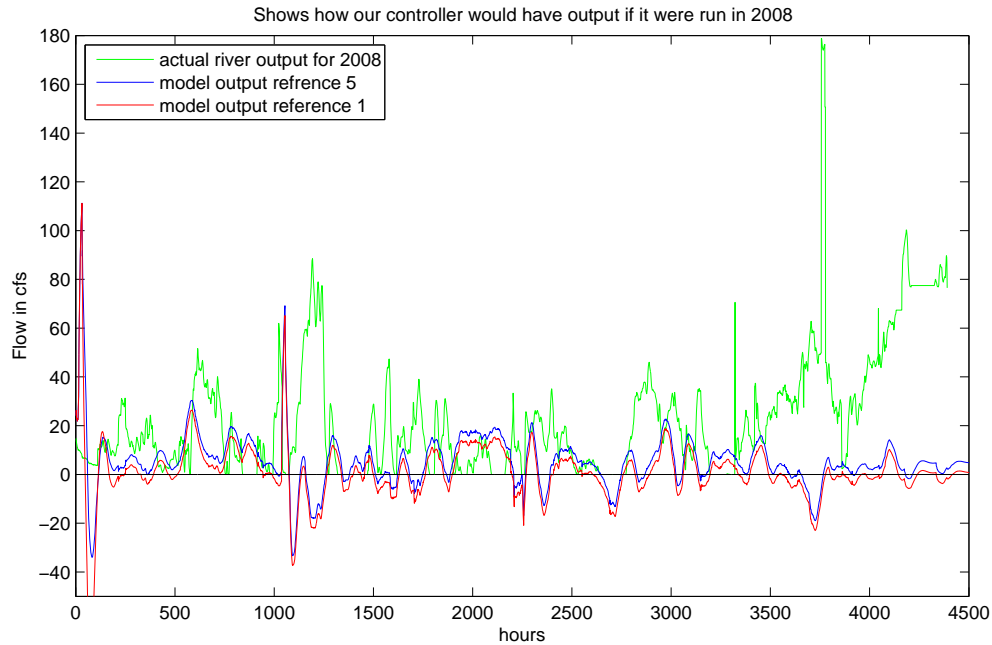


Figure 8. An estimate of how the river output would have been if the controller was running on the system in 2008 with a reference command of 1 cfs (red) and 5 cfs (blue). The green graph shows the actual river output.

Table 3. Water savings and shortages at different output targets (in acre feet)

Reference Command For Output	Water Saved	Water Shorted (Less than Zero)
$r = 1$ cfs	6514 AF	650 AF
$r = 5$ cfs	5508 AF	312 AF
$r = 10$ cfs	3985 AF	140 AF
$r = 15$ cfs	2375 AF	55 AF
$r = 20$ cfs	716 AF	20 AF

IMPLEMENTATION INTO EXISTING SCADA SYSTEM

In this section we describe how the control algorithm is implemented and added to the existing SCADA system. Until now the focus has been on creating a model of the river and designing a controller for the reservoir gate. First the existing SCADA system is described. Then, the three parts of the controller implementation are explained. Following the description of the system the two major problems of communication and security are discussed.

SCADA System

The preexisting system without the control algorithm consists of stations along the river, a server application called Loggernet to connect with the stations, a database for data storage, and a website for viewing the data. The data flow between these pieces is shown

by the red arrows in figure 9. Each station consists of a data-logger with measurement sensors, a solar powered battery charger, and a communication radio. Most stations also contain automated gates and a local controller of these gates so that the flow can be set by canal owners and the gates will automatically adjust to let that flow through the gate. Gate automation is done by measuring the flow a little downstream of the gate and adjusting the gate until the flow matches the desired set point.

Campbell Scientific's application, Loggernet, is used to remotely connect to the data-loggers [2]. It is set to connect to each station and collect the flow and other sensor data every hour. It stores this data in a large database. Loggernet also has the capability to allow a user to manually connect to a station in order for a user to change the flow set point. An extensive website (www.sevierriver.org) allows anyone to access the data stored in the database for each station. After locating the station, historical data can be viewed and downloaded.

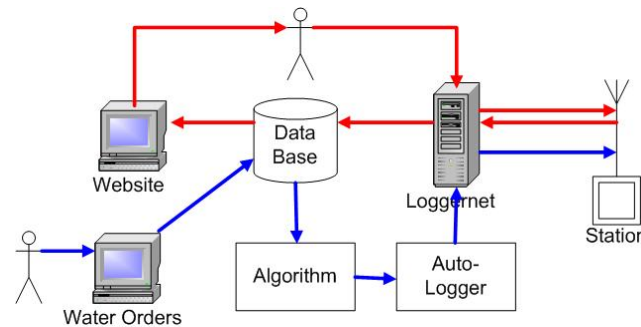


Figure 9. Data flow diagram of the original SCADA system (red), and with the addition of the controller (blue).

Control Algorithm Implementation

The architecture of the control algorithm contains three pieces which work together to control the reservoir and the gates on the canals. The data flow for the different pieces of the control implementation is shown by the blue arrows in figure 9. The first part is the online water order form. This form allows canal owners to go online and enter the future orders for their canals. These orders are stored in the database so they can be accessed by the controller.

The second part is the implementation of the controller itself. The controller has access to the database of water orders and current river flows. It calculates how much water to let out of the reservoir using the algorithm described in the previous section, and sends that value to the auto-logger. The controller also sends new set points for the canal to the auto-logger.

The third part of the implementation is the auto-logger which runs on top of Loggernet. This program receives gate set points from the control algorithm and uses Loggernet to communicate the set point to the data-logger. Just as in the original system Loggernet handles all of the radio communications. The reason for the auto-logger is to automate the sending of the set points to the station without a human initializing the change.

One major benefit of this three part implementation is its modularity. It allows the control algorithm to be completely redesigned or implemented in a new language without changing the rest of the implementation. Likewise, other parts may be redesigned or improved as well.

Communication with the Stations

Loggernet software handles the communication between the server and each of the stations on the river. This communication travels via ethernet to a data hut near Richfield, then from there travels by radio to the different stations. These radio communication channels present the single most challenging part of making the system functional. The radio channels can be very noisy and unreliable at times. It can take a long time to connect or transmit data and the signal can be dropped entirely. Also the entire system of stations is on just a couple of radio frequencies. If someone is using the frequency to change a stations flow or read its values then the platform will not be able to connect to any other station on the same frequency. This can become more serious when someone forgets to log off and keeps the line tied up for several hours.

The system has been designed to mitigate communications problems and increase the reliability of the system. The system connects to the stations once per hour to make needed changes, this reduces traffic on the radio frequencies. The system is also designed to make all the needed changes on the canals so there should not be a need for users to use the communication channels as often.

Before the system ties up the communication lines, it first checks to determine whether there has been a significant change in the order for the canal or reservoir release. A significant change is .25 cfs for a canal and 2 cfs for the reservoir. If there has not been a significant change, it will not call to make a change. The only communication initiated by the system is for those stations where there is a significant change. If communication fails, the system will quit all communications and wait for 30 seconds and then try again. After three separate tries it will give up until the next hour.

Security and Reliability

Because water is the livelihood of those who are served by the reservoir, great care must be taken to make the system secure and reliable. If the system fails, the failure must be recognized in time to minimize the damage. If a farmer loses water for a few days his crop for the year could be ruined. This danger causes many of the water users to be reluctant to allow the system to control the reservoir and the gates.

The online water order entry form uses SSL encryption and a password system to ensure the identity of those using the system. Those who use the system must have their account manually verified and be given access to each canal they are to control. Users are not allowed to see any canal that has not been assigned to them.

If at any time the system has trouble making a change on a canal or the reservoir, it will send an email notification that there is a problem. The email will then be forwarded via short message service (SMS) to the administrator's cell phone. That way, someone knows of the problem immediately, and can evaluate and fix the problem. Possible errors that are reported are communication errors, software errors, file read or syntax errors, or if the flow is outside of some pre-specified range.

As a last fail-safe, every station is equipped with a flag that must be turned on for the system to control the gate. The value, AUTOMODL, must be set to 1 otherwise the controller will not attempt to change the flow set point. If at any time the system must be discontinued for an emergency reason, one can simply change the flag to be zero. Having the fail-safe flag means that in the worst case scenario, the administrator can just turn off the controller and move the gate manually.

CONCLUSION

A fully automated system has been designed for the control of the Piute Dam and the eight canals downstream. The system takes water orders for anticipated demand from the canal operators and uses them to determine how much water should be released from the reservoir. A controller has been developed and validated to outperform previous control methods and has the potential to conserve water for years to come. The modular design of the system allows for the modification of the controller if needed. Several simulations have been run to validate the stability and potential water savings of the controller and it is planned to be run on the Sevier River system during the summer of 2010.

REFERENCES

- [1] J.-P. Baume M. Rijo X. Litrico, V. Fromion. Modeling and pi control of an irrigation canal. In *Proceedings of European Control Conference ECC03*, Cambridge, UK, 2003.
- [2] Campbell Scientific, Inc. *Loggernet Software Development Kit Programmer's Reference*, 2.2 edition.
- [3] Xavier Litrico. Robust imc flow control of simo dam-river open-channel systems. *IEEE Transactions on Control Systems Technology*, 10(3):432–437, 2002.
- [4] P.-O. Malaterre and J.-P. Baume. Modeling and regulation of irrigation canals: existing applications and ongoing researches. In *Proceedings of IEEE Conference on System, Man, and Cybernetics*, pages 3850–3855, San Diego, CA, USA, 1998.
- [5] M. Maxwell and S. Warnick. Modeling and identification of the sevier river system. In *Proceedings of American Control Conference ACC06*, 2006.
- [6] Erik Weyer. System identification of an open water channel. *Control Engineering Practice*, 9:1289–1299, 2001.

[7] Erik Weyer. Lq control of an irrigation channel. In *Proceedings of IEEE Conference on Decision and Control*, pages 750–755, Maui, Hawaii, USA, 2003.