TUNING ALGORITHMS
For Automated Canal Control

ARI Project Number: ARI 04-3-005; Foundation 49505

Project Title: Canal Automation Optimization

September 2005

by
Dr. Xianshu Piao and Dr. Charles Burt

Irrigation Training and Research Center (ITRC)
California Polytechnic State University (Cal Poly)
San Luis Obispo, CA 93407
805-756-2434
www.itrc.org
xpiao@calpoly.edu  cburt@calpoly.edu
Tuning Algorithms for Automated Canal Control


Irrigation Training and Research Center (ITRC) – www.itrc.org
# TABLE OF CONTENTS

## Preface

- Canal Automation ................................................................. 1
- Distributed Control Definition ............................................... 1
- Control Algorithms ............................................................... 1
- The Tuning Process .............................................................. 2
- Summary of this Report ........................................................ 6

## Chapter 1. Control Algorithms

- Different Types of Control ................................................... 7
- General Description of Control Types .................................... 7
- Particular Control Situations Encountered by ITRC ................. 7
- Selecting a Control Logic ...................................................... 10
- Two Control Logics Widely Recommended by ITRC ............... 10
- General Forms of Control Logic ............................................ 11
- Converting a Flow Rate Change to Movement ..................... 12
- Pump Speed ........................................................................ 12
- Gates .................................................................................. 13
- Heuristic Adjustments for the Initial Movement Calculation ..... 17
  1. The KP and KI of PIF constants adjustments based on flow condition ..................................... 17
  2. Anti-Hunt Function .......................................................... 17
  3. Step Function ................................................................... 18
  4. A Memory of Previous Gate Movement ......................... 18
- Feedforward Control ............................................................ 19
- Introduction ........................................................................ 19
  1. Base Case: Step Applied to Head Gate .............................. 19
  2. Step Applied to all Gates .................................................. 20
  3. Ramp Down Setpoint ........................................................ 21
  4. Ramp Down Step and Setpoint on all Gates ...................... 22
  5. Flow Control (Accurate) .................................................... 23
  6. Flow Control (Inaccurate) .................................................. 24
  Feedback and Feedforward Control from Last Pool ............. 26
  Programming the Control Algorithm for CanalCAD Simulation ......................................................... 28
- Summary ............................................................................ 28

## Chapter 2. CanalCAD Simulation Model Software

- General Description ............................................................ 30
- Actual vs. Modeled Results .................................................. 30
- Discrepancies with CanalCAD Results ................................. 31

## Chapter 3. Characterizing the Pools

- Surface Area ($A_s$) and Delay Time ($\tau$) ........................... 34
- Introduction to $A_s$ .............................................................. 34
- Introduction to $\tau$ .............................................................. 36
Calculating $A_s$ and Tau ....................................................................................................................... 37
Resonance ........................................................................................................................................... 40
Ground Frequency ($\omega_r$) and its Resonance Peak ($R_p$) ................................................................. 40
Effects of the Physical Properties of the Pool on Resonance .............................................................. 41
Critical Frequency ($\omega_{cr}$) and its Critical Resonance Peak ($R_{cr}$) ............................................... 43
Using a Small Computational Time Step ............................................................................................ 46
Summary Points ................................................................................................................................... 49

Chapter 4. Optimizing the Control Logic Parameters ........................................................................ 50
Background .......................................................................................................................................... 50
How MatLab Optimization Works ...................................................................................................... 50
Decoupler Optimization inside the MatLab Routines ........................................................................ 53
Improving Unsatisfactory Results ....................................................................................................... 57
Effects of Changing the $R_{cr}$ and $A_s$ Input Values ........................................................................... 57
Manually Changing FC to Improve Gate Movement .......................................................................... 58
Resetting FE2 as Zero .......................................................................................................................... 59
Using a FE2 Scaler to Improve Upstream Control Effects .................................................................. 60
Tuning Summary .................................................................................................................................. 61

Chapter 5. Recommendations and Limitations ............................................................................. 64
Algorithm Choice: Reduction Factor Control with Replogle Flumes .............................................. 64
Special Considerations for Algorithm Tuning .................................................................................... 69
Limitations to the Vertical Gate Opening ............................................................................................ 69
Use the Gate Accumulator to Zero-In On the Water Level to Target ............................................... 73
Using a Bigger Ramp Time for Smooth Response to Big Turnout Flow Changes .............................. 75
Using Control Based on the Water Surface Elevation for Pools with a Variable Invert Bottom Slope .................................................................................................................................. 76
How to Move Multiple Parallel On-Site Gates .................................................................................... 78
Considerations for Working with Equipment and Integrators ............................................................. 79
PLC and Sensor Constraints .................................................................................................................. 79
PLC Programming by Integrators ........................................................................................................ 81
Control Code Programming in ISaGRAF ............................................................................................. 83
Overview of ITRC Control Algorithm and ISaGRAF Modules of Flow Chart in Structured Text Language .................................................................................................................................. 84

References ............................................................................................................................................ 86
LIST OF FIGURES

Figure 1. A “big picture” look at the automation process for gates and/or pumps ......................... 3
Figure 2. PI or PIF tuning control process using CanalCAD and MatLab ................................... 4
Figure 3. An overview of the ITRC control algorithm ................................................................. 5
Figure 1-1. Head loss (ΔH) depends on the current gate position ............................................... 14
Figure 1-2. Longitudinal view of Delta Mendota Canal ............................................................. 19
Figure 1-3. Flow change made at the head (Check 13) reaches the last check (21) about 6 hours later ................................................................................................................................. 20
Figure 1-4. Flow change made at the same time from head (Check 13) to last check (Check 21) .... 21
Figure 1-5. Ramp down level setpoint for all gates to speed up delivery ................................... 22
Figure 1-6. Ramp down level setpoint and step on the flow for all gates to speed up delivery ...... 22
Figure 1-7. Flow through the gates during flow control ............................................................. 23
Figure 1-8. Water level upstream of the gates during flow control ............................................. 23
Figure 1-9. Gate opening of the gates during flow control ........................................................... 24
Figure 1-10. Flow through the gates when performing flow control with varied ±20% coefficient ... 25
Figure 1-11. Water levels upstream of gates when performing flow control with varied ±20% coefficient ......................................................................................................................... 25
Figure 1-12. Gates openings when performing flow control with varied ±20% coefficient ........... 26
Figure 1-13. Flow through the gates with (Feedforward + Feedback) from reservoir to head gate with all other gates maintaining upstream water level control ......................................... 27
Figure 1-14. Water level drops at reservoir and upstream of each gate with (Feedforward + Feedback) from reservoir to head gate with all other gates maintaining upstream water level control ................................................................. 27
Figure 1-15. Gate openings with (Feedforward + Feedback) from reservoir to head gate with all other gates maintaining upstream water level control .............................................. 28
Figure 2-1. The duplicated harmonic flow and improved flow effects for Pump 1 at Patterson Canal ........................................................................................................................................ 31
Figure 2-2. The small bug regarding the MTx3 elevation in CanalCAD ........................................... 32
Figure 2-3. Simulated and actual water level changes at Portuguese Bend Canal, Sutter Mutual Water Company ........................................................................................................................... 32
Figure 3-1. The target water level (and its corresponding surface area) varies according to flow rate ........................................................................................................................................... 35
Figure 3-2. Water level increase over time, resulting from a 55 CFS (5% of max flow) inflow increase and a constant inflow (immediate upstream control) .................................................. 38
Figure 3-3. The tradeoff between pool properties and resonance ................................................... 42
Figure 3-4. KPα beginning with unacceptable gate oscillations ................................................... 45
Figure 3-5. The difference in the resonance peak determined by using 1-sec, 5-sec, 1-min computational time steps ................................................................................................................... 47
Figure 3-6. Downstream control simulation results from the initial tuning of PI algorithms on the USBF Canal using a 1-minute simulation time step and a 1-minute control time step .... 48
Figure 3-7. Downstream control simulation results on the USBF Canal using a 1-second simulation time step and a 1-minute control time step, still using the PI algorithm .... 48
Figure 4-1. Sketch of the decoupler principle ............................................................................. 54
Figure 4-2. Canal controlled with optimized PI/PIF controllers (no decouplers) using MatLab TUNE (January 2004 version) ................................................................. 55
Figure 4-3. Canal controlled with Unoptimized PI/PIF controllers, extended with decouplers (kd=1) ............................................................................................................................... 55
Figure 4-4. Canal controlled with optimized PI/PIF controllers, extended with decouplers ......... 56
Figure 4-5. Control effect comparison for CCIDLMC Checks 3 & 4 using the original FC and an increased FC at low flow conditions ............................................................................................ 58

Figure 4-6. The improved control effects of introducing an FE2 scaler for Gila Gravity “Y” check upstream control ........................................................................................................... 61

Figure 5-1. Simulated effects of using PI control at CRIT Check 56 .............................................. 63

Figure 5-2. Simulated effects of using Reduction Factor control with a RF=1 at CRIT Check 56 ....... 65

Figure 5-3. Simulated effects of using Reduction Factor control with a RF=0.28 at Check 56 .......... 66

Figure 5-4. The control effect of using Reduction Factor with the “a” and “b” adjustments .......... 67

Figure 5-6. Gate control problem IS NOT observed for the 10 sec. computational time step simulation when the gate must stay 0.1’ below the upstream water surface elevation .... 71

Figure 5-7. Gate control problem IS observed for the 1 sec. computational time step simulation when the gate must stay 0.1’ below the upstream water surface elevation .................. 71

Figure 5-8. Elimination of gate control problem for the 1 sec. computational time step simulation when the gate must stay 0.4’ below the upstream water surface elevation – the correct limitation .......................................................................................................................... 72

Figure 5-9. Comparison of water level control at DMC’s Gate 3 using GA and without using GA ... 74

Figure 5-10. When $|GA| \geq MnMG$, gate is moved by GA, gate control is unsatisfactory .......... 74

Figure 5-11. When $|GA| \geq 2 \times MnMG$, gate is moved by $\frac{1}{2} \times GA$ – results in improved gate control .............................................................................................................................. 75

Figure 5-12. Comparison of water level response at Sutter Mutual’s Portuguese Bend using ramp time = 2 minutes and ramp time = $|flow \ change| \times (5 \ minutes / 10 \ CFS \ flow \ change)$ .. 76

Figure 5-13. The Patterson Canal invert profile for Pool#1, where there is a slope variation ........ 77

Figure 5-14. Patterson pump #1 control effect comparison between those based on water depth and on water surface elevation .................................................................................... 78

Figure 5-15. ADFM and filtered signals at Headgate Rock Dam – CRIT irrigation project, Arizona .......................................................................................................................... 80

Figure 5-16. Relationship between ISaGRAF and the PLC ............................................................ 83

Figure 5-17. The ISaGRAF modules for a typical canal control .................................................... 84

Figure 5-18. The ISaGRAF programming in flow chart with structured text language ............... 84
# LIST OF ABBREVIATIONS AND SYMBOLS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_s$</td>
<td>Surface Area</td>
</tr>
<tr>
<td>ADFM</td>
<td>Acoustic Data Flow Meter</td>
</tr>
<tr>
<td>CCID</td>
<td>Central California Irrigation District</td>
</tr>
<tr>
<td>CFS</td>
<td>Cubic Feet Per Second</td>
</tr>
<tr>
<td>CRIT</td>
<td>Colorado River Indian Tribes</td>
</tr>
<tr>
<td>DMC</td>
<td>Delta Mendota Canal</td>
</tr>
<tr>
<td>d/s</td>
<td>downstream</td>
</tr>
<tr>
<td>DU</td>
<td>Required Gate Movement</td>
</tr>
<tr>
<td>ENOW</td>
<td>Present Unfiltered Error</td>
</tr>
<tr>
<td>FC</td>
<td>Filter Constant</td>
</tr>
<tr>
<td>FE</td>
<td>Filtered Error</td>
</tr>
<tr>
<td>GA</td>
<td>Gate Movement Accumulator</td>
</tr>
<tr>
<td>GN</td>
<td>Number of Gates</td>
</tr>
<tr>
<td>GW</td>
<td>Gate Width</td>
</tr>
<tr>
<td>H</td>
<td>Head</td>
</tr>
<tr>
<td>$H_t$</td>
<td>Water Level Height</td>
</tr>
<tr>
<td>ITRC</td>
<td>Irrigation Training and Research Center</td>
</tr>
<tr>
<td>KI</td>
<td>Integral constant for filtered water level error</td>
</tr>
<tr>
<td>KP</td>
<td>Proportional constant for filtered water level error difference</td>
</tr>
<tr>
<td>KP$_{cr}$</td>
<td>Critical Proportional Constant</td>
</tr>
<tr>
<td>L</td>
<td>Length of the pool</td>
</tr>
<tr>
<td>MnMG</td>
<td>Minimum Gate Movement</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional-Integral</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional-Integral-Derivative</td>
</tr>
<tr>
<td>PIF</td>
<td>Proportional-Integral-Filter</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>Q</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>QNOW</td>
<td>Present Measured Flow</td>
</tr>
<tr>
<td>QT</td>
<td>Target Flow Rate</td>
</tr>
<tr>
<td>R$_{cr}$</td>
<td>Critical Resonance Peak</td>
</tr>
<tr>
<td>R$_p$</td>
<td>Resonance Peak</td>
</tr>
<tr>
<td>RF</td>
<td>Reduction Factor</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>$\tau_d$</td>
<td>Delay Time</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Control Time Step</td>
</tr>
<tr>
<td>TDH</td>
<td>Total Dynamic Head</td>
</tr>
<tr>
<td>U</td>
<td>Vertical weir height above the gate floor</td>
</tr>
<tr>
<td>u/s</td>
<td>upstream</td>
</tr>
<tr>
<td>USBFC</td>
<td>Umatilla Stanfield-Branch Furnish Canal</td>
</tr>
<tr>
<td>V</td>
<td>Downstream flow speed of a wave</td>
</tr>
<tr>
<td>VFD</td>
<td>Variable Frequency Drive</td>
</tr>
<tr>
<td>VGO</td>
<td>Vertical Gate Opening</td>
</tr>
<tr>
<td>W</td>
<td>Width at top of a pool</td>
</tr>
<tr>
<td>Y</td>
<td>Depth</td>
</tr>
<tr>
<td>$\omega_{cr}$</td>
<td>Critical Frequency</td>
</tr>
<tr>
<td>$\omega_r$</td>
<td>Ground Frequency</td>
</tr>
</tbody>
</table>
Canal Automation

Canal automation refers to closed-loop control in which a gate or pump changes its position or running speed in response to a measured water level, flow rate, or pressure because that level, rate, or pressure is different from the intended target value. "Closed loop" means that the action is performed without any human intervention. The automation may be performed through hydraulics, electronics, or a combination of these means (Burt and Piao, 2002).

Distributed Control Definition

All ITRC canal automation work is for distributed control. Distributed work utilizes a PLC at each control structure check or pump. The PLCs operate automatically and independently (one per site), but they are remotely monitored via a Supervisory Control and Data Acquisition (SCADA) system at the irrigation district office, through radio communications in most cases. From the office, a person can switch any device to "manual" operation, or can change target water levels, pressures, or flow rates.

Control Algorithms

For canal gate and/or pumping automation to be successful, there must be a control algorithm (formula) inside each PLC, which computes the required change in gate position or pump speed (in the case of Variable Frequency Drive [VFD]-controlled pumps). The simple explanation is that the required change in gate position or pump speed will depend upon the deviation of the actual (measured) water level or flow rate from the target value. In reality, the change in gate position or pump speed is much more complicated and depends on several factors.

There are several formula algorithms, or variations of formulas, used by ITRC for gate and pump control. The first things to consider are selecting the proper type of control desired, and the proper formula to use. The algorithms contain several constants. These constants will have different values depending upon two things: (1) the gate or pump characteristics, and (2) the canal pool dimensions. Once the type of control and formula have been chosen, the control algorithm related constants must be “tuned”. “Tuning” involves determining the proper values of the constants to achieve a rapid yet stable and robust control.

In general, once the control algorithm has been tuned and implemented in the PLC, automatic control proceeds as follows:

1. Based on the deviation of the measured value from the target value, the control algorithm computes the required change in flow rate that is needed.
2. Based on the gate or pump characteristics, it is determined how much the gate should be moved or the pump speed should be changed to achieve the desired flow rate change.

3. For gate control, the positions of several on-site parallel gates determine which gate should be lowered or raised. For pump control, the running speeds of the VFDs and/or the running states of regular pumps determine which VFD’s speed should be changed and by what amount, and/or which of the regular pumps should be turned on or off.

**The Tuning Process**

The key to reliable control is a properly tuned control algorithm. The process for tuning a control algorithm for canal automation, simplistically illustrated in Figure 1 below, is quite complex. It involves the use of CanalCAD simulation software and an optimization code in MatLab. CanalCAD is used at the beginning of the process to characterize the pools in the canal. This information is then used in MatLab to determine the optimized constants (or “parameters”) for each pool’s control algorithm. A CanalCAD simulation is then run to troubleshoot the control effects using the optimized parameters.

If the new control effects are unsatisfactory, the process of optimizing the parameters and simulating the control effects are repeated until satisfactory control effects are achieved. Control effects are deemed satisfactory or unsatisfactory based on

1. the accuracy of the actual water level control as compared to the target value, and
2. the gate or pump response – the amount of pump/gate overshoot and how long it takes the pump or gate to stabilize in a steady state.
Figure 1. A “big picture” look at the automation process for gates and/or pumps

Through many simulations and actual field implementations, ITRC has acquired new understanding of problems and solutions within each step of the tuning process. A more detailed view of the process illustrated above is presented in Figure 2.
Figure 2. PI or PIF tuning control process using CanalCAD and MatLab

In Figure 2, the two sections circled in red are the core of the ITRC control algorithm. An overview of the ITRC control algorithm is shown in Figure 3.
PLC and some variable initialization

Scale Water level & gate position sensors

Manually select the primary sensor from the redundant sensors

Alarms inspections for the selected primary sensor comparing with actual high and low level or gate position values

Alarms inspections for the difference between the two redundant sensors

Alarms inspections for SCADA Intrusion, SCADA Input Battery, PLC Lithium Battery etc

Call the associated control scheme module of ITRC specific algorithm

i.e. the Upstream WL control, Downstream WL control, or Flow Control Module

Pick up one gate to move based on the philosophy that is to the right

Readjust the required gate movement before initiating the movement of the selected gate

Dynamic protection such as time limiter and the items to the right while the gate is moving

Always lower down the highest gate and raise the lowest gate only from the available ones among those on-site multiple parallel gates

The readjustments and dynamic protection are based on the hard & soft limit switches, radial gate submergence assurance as well as the overshot gate overshoot limits, etc

Figure 3. An overview of the ITRC control algorithm

In Figure 3, the block circled in red is the control equation; the other steps are necessary to ensure that the algorithm works properly.

The general overview shown in Figure 3 is edited for each situation in flow chart format as part of the control code in an ISaGRAF programming environment, inside which the integrator adds some modules to create the individual system’s communication, alarm dialing functions, etc. For details about ISaGRAF and its programming, refer to Chapter 5.
Summary of this Report

This report provides a clarification of the theoretical principles behind most of the steps in Figure 2, as well as a summary of what we have learned through trial and error, as we work to fine-tune the process. Briefly, these instructions are as follows:

1. Determine control algorithm (Chapter 1) and program in Fortran for CanalCAD simulation.
2. Build model in CanalCAD (Chapter 2)
   a. Enter pool dimensions, etc.
   b. Set Manning’s
   c. Export data
3. Determine MatLab inputs (pool characteristics) (Chapter 3)
   a. Aₜ and Tau
   b. K_Pcr (immediate downstream or upstream control)
   c. Delay time
4. Initial MatLab optimization (Chapter 4)
   a. Input variables into code
5. Run initial CanalCAD simulation (Chapter 4)
   a. Select Fortran control file
   b. Enter gate characteristics
   c. Input control constants into CanalCAD
6. Tune (Chapters 4 and 5)
   a. Change inputs in MatLab and re-optimize
   b. Manually change constants in CanalCAD
   c. Re-simulate in CanalCAD
CHAPTER 1. CONTROL ALGORITHMS

Different Types of Control

General Description of Control Types

Normally, canal automation controls either gates, pumps, or both. If gates are controlled, they are usually either radial gates or overshot gates, or some combination of the two. If pumps are controlled, they are either regular pumps or VFD (Variable Frequency Drive) pumps.

In any of these situations, control is based on the measured value of either water level or flow at a certain target point. If the control is based on water level, the measured target point can be located either immediately downstream or immediately upstream from the controlled structure (pump or gate), or at an intermediate location in the middle of the downstream pool. If the target point is located in the downstream pool, it could be the exact intermediate point or the bival-method point, a weighted point between the head point and the downstream end of the pool.

If the control is based on flow, the flow is usually measured over a Replogle flume or by using a flow meter located downstream of the control structure (gate or pump).

As can be seen, there are actually many types of canal control. The control for any particular location could be any one of the permutations and combinations of “gate or pump control, gate/pump type, water level or flow control, target point location” listed above.

Particular Control Situations Encountered by ITRC

Based on situations encountered in the field, ITRC has broken gate and pump control down into several definitive types, listed below with examples from irrigation districts. Although each situation will be different, the cases listed in the table below represent the majority of conditions encountered with distributed control – our preferred method of automation at this time.
<table>
<thead>
<tr>
<th>Type of Control</th>
<th>Description</th>
<th>Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Upstream Control</strong></td>
<td>Radial gates, no side flow</td>
<td>Delta-Mendota Canal</td>
<td>8 checks – study only</td>
</tr>
<tr>
<td></td>
<td>Radial gates, long weirs on side</td>
<td>Upper part of Government Highline Canal, Grand Junction, CO</td>
<td>9 checks – already implemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper part of Tehama-Colusa Canal</td>
<td>not yet implemented</td>
</tr>
<tr>
<td></td>
<td>Overshot gates</td>
<td>South San Joaquin ID, Manteca, CA</td>
<td>not yet implemented</td>
</tr>
<tr>
<td></td>
<td>Overshot or radial gates</td>
<td>CCID upper main canal, CA</td>
<td>to be implemented soon</td>
</tr>
<tr>
<td><strong>Downstream Control</strong></td>
<td>Five Gates, Reclamation District 108, Grimes, CA</td>
<td>Single gate – implemented immediately downstream</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Central California ID Lower Main Canal</td>
<td>Radial gates with side flashboards to maintain immediate d/s control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lower part of Tehama-Colusa Canal</td>
<td>Maintaining the water level that is 200’-3860’ d/s of current stilling well location</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Corning Canal</td>
<td>Far end downstream control of current stilling well location</td>
<td></td>
</tr>
<tr>
<td>Pumps</td>
<td>Patterson Irrigation District, Patterson, CA</td>
<td>5 pools automated, using 2 intermediate water levels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>West Stanislaus</td>
<td>4 stations, all automated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portuguese Bend, Sutter Mutual Water Co.</td>
<td>1 pump, intermediate level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overshot gate</td>
<td>Stanfield Branch-Furnish, Umatilla</td>
<td>1 radial, 3 overshot gates</td>
</tr>
<tr>
<td></td>
<td>Radial gate with intermediate bival d/s control</td>
<td>Berrenda Mesa Water District</td>
<td>13 radial structures with bival downstream control</td>
</tr>
</tbody>
</table>
### Flow Control

<table>
<thead>
<tr>
<th>Location</th>
<th>Control Method</th>
<th>Implementation Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower part of Government Highline Canal, Grand Junction, CO</td>
<td>4 checks – to be implemented soon with an automatic switch to change to upstream control if upstream level is beyond the normal range</td>
<td></td>
</tr>
<tr>
<td>Tulare ID, Watchuma Ditch</td>
<td>Sluice gate with about 200’ d/s water level control</td>
<td></td>
</tr>
<tr>
<td>Patterson 3 South Lateral</td>
<td>Combination of pumps and sluice gate</td>
<td></td>
</tr>
<tr>
<td>Central CA lower main Ingomar Reservoir</td>
<td>already implemented – control target location is immediately downstream of the reservoir outlet</td>
<td></td>
</tr>
<tr>
<td>Headgate Rock Dam, Colorado River Indian Tribe (CRIT)</td>
<td>Radial gate, with ADFM 36’ downstream to measure flow</td>
<td></td>
</tr>
<tr>
<td>Check 56, CRIT, Poston, AZ</td>
<td>Radial gate, with a Replogle Flume d/s to measure flow</td>
<td></td>
</tr>
<tr>
<td>The logic provides assurance that the u/s level is always within the normal range; it will switch automatically to maintain the u/s level until the user reselects the d/s flow control in HMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tulare ID, North Branch</td>
<td>Sluice gates (4 in parallel) with a Parshall flume d/s</td>
<td></td>
</tr>
<tr>
<td>Tulare ID, Creamline</td>
<td>Radial gate, with a Cipoletti weir d/s to measure flow</td>
<td></td>
</tr>
<tr>
<td>Klamath, OR</td>
<td>Radial gate, with a SonTek Side-Looking (SL) ADFM about 500’ d/s</td>
<td></td>
</tr>
<tr>
<td>already implemented</td>
<td>already implemented</td>
<td>already implemented</td>
</tr>
</tbody>
</table>
Selecting a Control Logic

Different situations call for different kinds of control logic. ITRC has historically used several kinds of control logic to meet different needs such as PI (Proportional-Integral) and PID (Proportional-Integral-Derivative) as well as Filter-only control logics, among others. The most widely recommended logics in ITRC canal automation projects are PIF (Proportional-Integral-Filter) and Reduction Factor (RF) logic, which can generally solve all control situations. (The PID, PI, Filter logic, etc. were illustrated in Summary Report Version Sept. 2002; this report only illustrates the latest that is currently being used in ITRC control algorithms.)

Two Control Logics Widely Recommended by ITRC

Reduction Factor (RF) Control

RF control is flow control (through a radial gate or sluice gates) based on one of five things:
1. the water level on a Replogle Flume,
2. head over a Parshall flume,
3. head over a Cipoletti weir,
4. the gate flow that is calculated using the head equation, or
5. the flow measured by an electric flow meter.

If the flow is measured by an electric flow meter, then ITRC provides backup measurements by using the head equation to calculate the flow through or over the gates, which is vital for the robustness of the control when the power of the electrical meter is down or it is taken out for servicing.

The Reduction Factor logic provides a solid control for the check structure, which normally performs a different control scheme than its upstream or downstream check, if there is one. For example, CRIT Check 56 performs its flow control based on a Replogle flume, but its Check 42 upstream and Check 70 downstream perform upstream water level control. The upstream water level flow control is also in danger of falling out of normal range when the gate tries to maintain the downstream flow, if there are no other maintaining devices. This situation occurred at CRIT Check 56; the solution devised by ITRC provided an automatic switch from downstream flow control to upstream water level control if the upstream level falls out the normal range. A similar flow control check structure at Camp 7 in Highline Canal, Grand Junction, CO, will not have this problem because its upstream level is maintained by pumps and a Langemann gate.

Proportional-Integral-Filter (PIF) control

In pools with severe resonance problems (such as the Stanfield Branch-Furnish Canal near Hermiston, Oregon, which was modeled for immediate downstream control), it is important to use a “filter” in the PI equation. If a filter is not used, it is almost impossible to avoid instability with 1 minute or greater control time steps. The filter accounts for the previous error, and all previous errors, but as the errors become more distant in time, their influence is lessened.
There have been several cases in which the PIF control logic was used:

- Gate position control based on the water depth:
  - immediately downstream of gates,
  - intermediately downstream of gates,
  - immediately upstream of gates, or
  - at the far downstream end of gates.

- Pumping plant flow rate control based on the estimated water depth:
  - downstream of the pumps at an intermediate location in the pool, or
  - at about an 82% intermediate location in the pool, or
  - immediately downstream of the pumping plant.

- Pump and gate control based on a far-end (3.3 miles downstream) water level with an upstream control gate in the middle (lower part of CCID canal, Ingomar Reservoir).

**General Forms of Control Logic**

It should be noted that the Reduction Factor always consists of positive values ranging from 0 to 1. The +/- values of the initial PIF constants that were optimized from the ITRC MatLab optimization routines come out with positive values for upstream control, and negative values for downstream control. Overshot and undershot gate types are not identified as positive or negative in the PIF constants; they are indicated by whether the gate dimension related constants are positive or negative. The equations for the Reduction Factor and PIF control logic are listed and described below.

**Reduction Factor Control (RF)**

The formula for Reduction Factor (RF) is given below:

\[ \Delta Q(k) = RF \times (QT - QNOW) \]

Where

- \( \Delta Q(k) \) = the change in flow rate through or over the gate or from the pump, in CFS. “k” is a natural number representing the \( k^{th} \) control action.
- RF = the dimensionless small reduction factor for the flow rate error at the present time step
- QT = the target flow rate at the present time step, set by the user, in CFS
- QNOW = the present measured flow rate, in CFS
- QNOW can/may be an equation using the head target that is set by the user for a Replogle flume or Cipoletti weir or Parshall flume, etc.
Proportional-Integral-Filter (PIF)
The new form of the equation is called the “PIF” equation. It can be described as follows:

\[
\Delta Q = (KP \times (FE1 - FE2)) + (KI \times FE1)
\]

(1-2)

Where

\[
\Delta Q = \text{change in flow rate through the gate, in } m^3/\text{sec}
\]

KP = the proportional constant for the filtered water level error difference between the present time step and previous time step

KI = the integral constant for the filtered water level error at the present time step

FE1 = the filtered error for the present time step

FE1 = (FC \times FE2) + [(1 – FC) \times ENOW]

where

FC = the filter constant

FE2 = the value of FE1 for the previous time step

ENOW = present unfiltered error

= (Actual water level – Target water level)

where

Actual water level is the average of the measurements taken since the previous control action.

Converting a Flow Rate Change to Movement

So far, the equations of RF and PIF control logics have all been given as flow rate changes. However, to actually apply this desired change as a form of control, these values must be incorporated into an overall control algorithm, which can calculate a change in pump speed or gate opening for radial gates or gate height for overshot gates.

The pump speed conversion from the computed flow change is based on if the pump’s Total Dynamic Head (TDH) changes; and the conversion for gate position from the computed flow change differs depending on the type of gate (radial gate, crest weir, check structure using a combination of gates, etc).

Pump Speed

To affect a desired flow rate change at a pumping plant, the desired flow rate change needs to be converted to a speed change for the VFD (Variable Frequency Drive) pumps.

For Constant TDH

This conversion is calculated based on the average percent speed change that is required for the VFD to increase the flow rate by 1 CFS (percent per CFS), which can be characterized as
\[
\frac{\Delta \text{Speed}}{\Delta Q} = f(\text{speed})
\]
along the whole range of the flow. This average must be calculated separately for each VFD pump.

To provide the most accurate control possible, the desired flow rate change must be considered very precisely. For example, at Portuguese Bend, where the pumps require a 1.3% increase in speed per CFS on average, the desired flow rate change needs to be calculated within tenths of a percent) to achieve an accurate enough pump speed change.

**For Changeable TDH**

If the TDH changes over time, it is a problem to have the constant speed change for a required flow change at different pump speeds. In this case the flow rate is needed for each individual pump when it is running at a certain speed. The flow rate for each pump can be derived from the flow rate meter readings, or based on a flow derivative equation based on the pump speed. The latter option is more accurate when taking into consideration calibration errors, defects and malfunctions, or power failures that may occur with the flow meter. The speed change corresponding to a required flow change can be expressed as a polynomial equation or as a set of linear equations for difference flow rate ranges.

There are two other situations in which the pump should not be kept running:

1. If the motor is cooled by a fan, the motor will overheat if the speed < about 50%.
2. Never let Q get to zero; if Q < 0.05 Q_{\text{max}}, then shut it off.

**Gates**

As mentioned earlier, to actually apply a desired flow rate change, the control algorithm must result in a movement of the gate. Flow through a gate is controlled by raising or lowering the gate. Since flow of the water through the gate is determined by the position of the gate, each gate has an equation that defines the relationship between gate position and flow rate. By combining this equation with a control logic equation, the ultimate control algorithm is determined. Determining how much to move the gate largely depends on how the big the desired flow rate change is.

**Gate Movement for Undershot Gates**

This can be seen in the following equation, which is used to calculate the required change in vertical gate opening for undershot gates such as a radial gate or sluice gate:

\[\Delta VGO = \frac{\Delta Q}{C_d \times GW \times \sqrt{2 \times g \times \Delta H}}\]  (1-3)

Where:

- \(\Delta VGO\) is the required change in vertical gate opening for one gate.
- \(\Delta Q\) is the desired flow rate change through the radial gates
C_d is a factor that accounts for the gate discharge smoothness, normally C_d = 0.65
GW is the gate width (for each gate)
g is the gravitational constant, g = 32.2
ΔH is the head loss, or difference between water levels on the upstream and downstream sides of the check structure

Note: The ITRC control logic always moves one gate at a time. If several on-site gates were moved at the same time, then (ΔVGO/GN) would be the required change in vertical gate opening for all gates, where GN is the number of parallel gates that are on-site and being controlled. There are two shortcomings for moving multiple gates at a time: one is the moving accuracy since there is an unavoidable error between the calculated gate target position and the where the gate actually stops; the other is the more rapid failure of the gate mechanics such as brakes.

**Head loss due to gate position**

In the equation above, it is apparent that the largest influence on the required change in gate position is the desired flow rate change. However, it is not the only influence. Although GW, GN, C_d and g are constants for each check structure, ΔH is not. As can be seen in the following figure, the value for ΔH (head loss) depends on the position of the gate.

![Figure 1-1. Head loss (ΔH) depends on the current gate position](image)

The above figure illustrates the inverse relationship between gate position and head loss (ΔH). If the gate is only slightly open (Opening 1), the head loss is large (ΔH_1). If the gate is wide open (Opening 2), the head loss is small (ΔH_2).
Since \( \Delta H \) varies according to gate position, the change in gate position needed to achieve a certain flow rate change also varies according to the current position of the gate. Say that it is desired to increase the flow rate by 1 CFS. Because \( \Delta VGO \) is inversely proportional to \( \Delta H \), if the gate is initially only slightly open, the head loss \( (\Delta H_1) \) is large, so the \( \Delta VGO \) required to change the flow rate by 1 CFS would be small. However, if the initial position of the gate is wide open, the head loss \( (\Delta H_2) \) is small, so the \( \Delta VGO \) required to change the flow rate by 1 CFS would be larger.

A simple explanation of this effect is that a larger head loss results in build-up of more water pressure upstream of the gate. This essentially results in an acceleration of the water’s velocity due to the force of gravity. Since the water is moving faster, only a slight change in the gate position is needed to achieve a flow rate change. However, when the water is moving slower (when the gate is wide open and the head loss small), the gate will need to be moved more to achieve the same flow rate change.

The desired control logic equation is substituted in for \( \Delta Q \) to get the final control algorithm.

1. **Final gate movement equation for PIF control logic.**

   If PIF control logic were used, the control algorithm would be:

   \[
   \Delta VGO = \frac{\left(KP \times (FE1 - FE2)\right) + \left(KI \times FE1\right)}{C_d \times (GN \times GW) \times \sqrt{2 \times g \times \Delta H}}
   \]  

   \( (1-4) \)

2. **If the Reduction Factor control logic is used, the final algorithm is as follows:**

   \[
   \Delta VGO = \frac{RF \times (QT - QNOW)}{C_d \times (GN \times GW) \times \sqrt{2 \times g \times \Delta H}} = \frac{QNOW \times RF \times (QT - QNOW)}{QNOW \times \sqrt{2 \times g \times \Delta H}} = Gate \ Position \times \frac{RF \times QENOW}{QNOW}
   \]

   \( (1-5) \)

In the final equation using the Reduction Factor, the water surface difference between the areas upstream and downstream of the radial gates does not need to be calculated.

**Overshot Crest Weirs**

The flow rate over the overshot gate is computed differently based on the free flow conditions and the submerged (flooded) flow conditions inside CanalCAD (a tool used by ITRC for canal control simulation). The computations are:
1. **Free-flowing case:** \((Y_{ds} - U) \geq \frac{2}{3} (Y_{us} - U)\)

The flow rate over a crest weir in a free flow condition can be expressed with equation 1-6 (consistent with CanalCAD manual Page 36):

\[
Q = \frac{2}{3} \sqrt[3]{\frac{1}{3} \times GW \times GN \times \sqrt{2 \times 32.2} \times (Y_u / s - U)^{1.5}} = 3.09 \times (Y_u / s - U)^{1.5}
\] (1-6)

Where:
- \(Q\) is the flow rate over the crest weirs in CFS
- \(U\) is the vertical weir height above the upstream canal invert in feet (it is assumed that all on-site weirs have the same height)
- \(Yu/s\) is upstream depth above the invert (it is assumed that the upstream invert is the same as the gate bottom)
- \(GN\) is the weir number in the control
- \(GW\) the effective gate length in feet

For the required flow change, the equation can be changed to resemble:

\[
\Delta Q = 3.09 \times GW \times 1.5 \times (Y_u / s - U)^{0.5} \times (-\Delta U)
\] (1-7)

\[
\Delta U = \frac{-\Delta Q}{3.09 \times GW \times 1.5 \times \sqrt{(Y_u / s - U)}}
\] (1-8)

If the PIF control logic is used, then the final equation of the gate movement for the overshot gate in free flow condition is:

\[
\Delta U = \frac{(KP \times (FE1 - FE2)) + (KI \times FE1)}{(-3.09 \times GW \times 1.5) \times \sqrt{(Y_u / s - U)}}
\] (1-9)

2. **Flooded (Submerged flow) case:** \((Y_{ds} - U) < \frac{2}{3} (Y_{us} - U)\)

The flow in the submerged (flooded) flow conditions can be expressed as:

\[
Q = GW \times \sqrt{2 \times 32.2} \times (Y_{ds} / s - U) \times \sqrt{Y_u / s - Y_{ds}}
\] (1-10)

The required flow change \(\Delta Q = 8.02 \times GW \times (-\Delta U) \times \sqrt{Y_u / s - Y_{ds}}\)

\[
\Delta U = \frac{-\Delta Q}{8.02 \times GW \times 1.5 \times \sqrt{Y_u / s - Y_{ds}}}
\] (1-12)

If the PIF control logic is used, then the final equation of the gate movement for the overshot gate in submerged flow condition is:
\[ \Delta U = \frac{(KP \times (FE1 - FE2)) + (KI \times FE1)}{(-8.02 \times GW \times 1.5) \times \sqrt{(Y_{ds} - U) \times \sqrt{Y_{u/s} - Y_{d/s}}}} \]  

(1-13)

The negative sign of the divider in equations (1-9) and (1-13) indicate that the gate is an overshot gate.

**Heuristic Adjustments for the Initial Movement Calculation**

A series of heuristic adjustments has been designed and used together with the PIF control logic, which enable the ITRC-provided logic and constants to have stable control and are suitable for all flow conditions. These adjustments are indispensable, and are basically comprised of the following facets:

1. **The KP and KI of PIF constants adjustments based on flow condition**

   It is normal that one KP/KI set of PIF constants work well for one flow condition, but not as well for other flow conditions. To solve this, the flow condition is determined before KP and KI are used in the PIF calculation.

   For undershot gates, the flow condition is decided by the quotient of (average of all vertical gate openings)/(upstream water depth). Ideally, the quotient should always be smaller than 1 since the ITRC logic is always trying to keep the undershot gate fully submerged. So, when the quotient is lower than a certain percentage such as 10-15%, the flow is regarded as low.

   For overshot gates, the flow condition is decided by the quotient between (average of all vertical gate heights)/(upstream water depth). Similarly, the ITRC logic always prevents the crest of the overshot weir from going past a certain distance, such as 0.2’ above the upstream water surface. When this quotient is larger than a certain percentage such as 80-90%, then the flow condition is regarded as low.

   The vertical gate heights or openings and the upstream depth mentioned above should be referenced by the same bottom elevation. In low flow conditions, the KP and KI were readjusted by a factor such as 0.3 or 0.5 times the given KP and KI, which most of the time is quite good for most conditions, ranging from low to high flows.

   These KP and KI adjustments based on the flow condition eliminate the occurrences of instabilities that are mostly found under low or very low flow conditions.

2. **Anti-Hunt Function**

   When only using a PI or PID logic which has no filter, the gate movement direction may appear correct when considering the error direction in each time step, but there is the danger that the gate may move too much, causing a compensating movement in the other direction.
during the following time step. If this happens, it is said that the gate is “hunting,” and waves are produced.

The effect of the filter in the PIF logic is to take out the impacts of measurement error and dampen overly aggressive gate movements. This does not thoroughly ensure that the filter will always give correct gate movements in each control time step, however. This effect can be described as follows:

When the level error is large, such as 0.15’, but the calculated gate movement is 0 or the calculated gate movement indicates a gate movement in the wrong direction.

In this case, the anti-hunt function is used, which means the filtered error is lowered for the previous time step (FE2) and the PIF equation is recalculated.

3. Step Function

Before ITRC began designing the KP/KI adjustments based on the flow condition, it was found that when there is a smaller error, such as 0.05’, the gate movement that was initially computed gives the correct direction but the amount is too large, such as 0.4’. In this case, this step function is applied:

If the error is smaller than a certain value input by the operator, and the initially calculated gate movement is bigger than a certain gate movement, then the PIF is recalculated using a stepped KP (such as 20 to 50 percent of KP).

This step function and the KP/KI adjustments based on flow conditions help to create moderate gate movement at all flow conditions.

4. A Memory of Previous Gate Movement

In a good control, the gate will never go up by more than a certain amount in each time step or go down by more than a certain amount in the following control time step. If this were to happen, then the smoothness of the control would not be achieved and instability and waves would occur.

Based on this aspect of a good control, a “memory” of the gate movement that was done in the previous time step is recorded and always compared with the calculated gate movement in the current time step. This is another way to prevent the gate from hunting. This memory of gate movement from the previous time step is used as follows:

After the initial required gate movement (DU) is calculated in this time step, if the calculated result in this time step (DU) and the memory of gate movement in the previous time step (DUPrev.) have opposite signs (if one is positive and the other negative), and both of their absolute values are higher than 1/3 of the maximum gate movement (0.6’-0.8’), then the gate will “hunt” and the control will no longer be smooth. In this case, the gate movement must be recalculated by multiplying the calculated result (DU) by 1/3.
Feedforward Control

For canal automation, the type of control in which a check structure transmits the required flow change and/or vertical gate opening to the next check upstream is called “feedforward control”.

Introduction

The setup of a control system in which feedforward is applied can speed up the delivery of water to a downstream reservoir. The control system must be based on local upstream control. In addition to the feedback control actions, feedforward control can be added in on an incidental basis.

Various feedforward methods were tested inside the Sobek canal simulation software using the Delta Mendota Canal as a model. The following summarizes the results of the simulation testing and recommendations on which control methods have the potential to be implemented in practice. Figure 1-2 below shows a longitudinal view of the canal.

![Figure 1-2. Longitudinal view of Delta Mendota Canal](image)

1. Base Case: Step Applied to Head Gate

First, the base case is simulated. This is the typical immediate upstream control strategy, in which each check structure controls its upstream water level. The controller uses
Proportional Integral Filter control. The control parameters are found with the optimization program TUNE by filtering the critical resonance peak for local upstream control.

After one hour a step in flow rate from 40 m$^3$/s to 50 m$^3$/s is applied at the upstream head gate (Check 13). The simulation results shown in Figure 1-3 show that the additional flow reaches the last pool (that is supposedly connected to the reservoir) after 6 hours.

![Figure 1-3. Flow change made at the head (Check 13) reaches the last check (21) about 6 hours later](image)

2. **Step Applied to all Gates**

To speed up the delivery, the step described above is directly applied at all structures as a feedforward action. The flow rate change of 10 m$^3$/s is applied after one hour at all gates as a one-time action.

From the simulation results it can be seen that directly after the step the water levels upstream of the gates drop and that the PIF-controllers try to correct for this deviation. Even though there is a direct flow into the last pool at the moment of the step, the flow is only 45 m$^3$/s at 5 hours and reaches 50 m$^3$/s after 6 hours.
3. Ramp Down Setpoint

Another way of directly giving more water is to ramp down the setpoint with the slope

\[ i_r = -\frac{\Delta Q_{FF}}{A_s} \]

For example, if the calculated required flow change using feedforward is 10 CFS, and the storage area is 100,000 ft², then the set point is ramped down at the slop (speed) of 0.0001 per second. As the water level after the setpoint change is higher than the setpoint, the PIF-controller will start releasing water by opening the gate. The slope is calculated from the required flow rate change and the storage area of the pool that is being controlled. The duration of the ramping down is set to the time it takes the wave to travel from one end of the pool to the other, as after this time the flow rate change of the upstream gate will have reached the controlled water level.

From the simulation results it can be seen that after 3 hours the inflow into the last pool is 50 m³/s. The reason why this flow is not immediately 50 m³/s is that the error is filtered and therefore the control action is slow.
4. Ramp Down Step and Setpoint on all Gates

Next, a combination of the previous two methods was tested. This combination is a direct step in flow rate from 40 m$^3$/s to 50 m$^3$/s to all gates as a one-time feedforward action and a ramp down of the setpoint.

The simulation results show a direct inflow into the last pool of 48 m$^3$/s. This result indicates that this seems to be a promising method. Note, however, that the water levels are off setpoint and need to be ramped back over a certain period of time.
5. Flow Control (Accurate)

A more obvious method is to impose the required change in flow rate at all structures over a certain period. If the change in inflow is kept equal to the change in outflow the water levels should not change considerably. After the fixed flows period the PIF-controller is turned on again to bring the water levels back to the setpoint. As the flow rate changes, waves are introduced in the pools; the fixed flow period must be chosen to dampen out these waves to a certain extent. A period of three times the delay time is selected, as this is approximately the time for a wave to travel the length of the pool four times. The friction will have dampened out the wave to some extent by then.

The simulation results show a direct inflow of 50 m$^3$/s into the last pool during the fixed flow period and a flow close to that value when the PIF-controllers are bringing the water levels back to set point. This is a promising method to show the gate opening, water level and gate flow in a more distinctive way. These are given as the three figures below.
6. Flow Control (Inaccurate)

A potential problem with the fixed flow controller is that in most cases a flow cannot be imposed accurately. Due to inaccurate estimation of structure formulas, changing environmental factors like waste blocking the gate, or inaccurate water level measurements, flows may be imposed that are up to 20% of the required flow. To investigate the impact that this has on the flow controller, the previous simulation is repeated, but now with discharge coefficients that are varied plus and minus 20%.

It can be seen from the simulation results that the fixed flow controller still functions well, even under this large mismatch between the flow equation used in the flow controller and the equation of the actual gate. This is because the mismatch is already eliminated in the base flow by the feedback controller and is only present in the relative change in flow rate. The flow through the gate, the water level upstream of the gate and the gate openings are shown as the three consecutive figures below.
Discharge through check structures

Figure 1-10. Flow through the gates when performing flow control with varied ±20% coefficient

Water level deviation upstream of check structures

Figure 1-11. Water levels upstream of gates when performing flow control with varied ±20% coefficient
Feedback and Feedforward Control from Last Pool

Another way of looking at the control problem is to include the control of the water level of the reservoir in the control loop. This way, the system reacts directly and smoothly to any (small) deviation of the water level of the reservoir from the setpoint, rather than having to take a sudden large flow rate change to compensate for a large error in the reservoir. This can be achieved in various ways:

- Using a schedule in which the offtakes from the reservoir are recorded. This can be used as a feedforward signal to release water at the head gate beforehand.
- Applying distant downstream control with decouplers at all pools. One drawback is that this requires communication lines along each pool.
- Using centralized control methods like Model Predictive Control. One drawback is that this requires communication lines to and from a central operating room.
- Applying distant downstream control in one loop from the reservoir to the head gate. The water level in the reservoir is measured and compared with the set point. This error is then sent all the way to the controller of the head gate, where the control action is computed with a PI-controller. If this is combined with feedforward as mentioned in the first bullet, the water level in the reservoir can be maintained close to the setpoint or smaller reservoirs can be built. A drawback is of course the communication line from the reservoir to the head gate, but this is only a one-way signal and can be transmitted, for example, through a telephone line.

To illustrate the method described in the last bullet, the following simulation is done. The last pool is considered as the reservoir (area=320000 m²). The last gate opens over a period of 10 hours such that the outflow increases from 40 m³/s to 45 m³/s. This slow change is to simulate the flow out of the reservoir due to downstream demands. For that reason no sudden large step is used as applied in the previous tests. The change in flow rate through the
gate is measured and sent, together with the deviation of the water level from the set point, to the controller of the head gate. Here, the change in flow rate is applied as a feedforward signal together with the PI-control action on the deviation as a feedback signal. All other gates are still under local upstream control.

From the results, it can be seen that the water level in the last pool (considered as the reservoir) only drops 10 cm.

---

**Figure 1-13. Flow through the gates with (Feedforward + Feedback) from reservoir to head gate with all other gates maintaining upstream water level control**

**Figure 1-14. Water level drops at reservoir and upstream of each gate with (Feedforward + Feedback) from reservoir to head gate with all other gates maintaining upstream water level control**
Tuning Algorithms for Automated Canal Control

Gate opening of check structures
step at Time = 1 hour

Current control constants
Gate opening (m)

Feedback and
Feedforward from
last pool

Figure 1-15. Gate openings with (Feedforward + Feedback) from reservoir to head gate with all other
gates maintaining upstream water level control

Programing the Control Algorithm for CanalCAD
Simulation

After the proper logic is selected, then it and the associated heuristic adjustments need to be
programmed into a separate subroutine in Fortran Language for CanalCAD simulation. The
Fortran version currently used by ITRC is Visual Fortran 6.5. The canal that will be
simulated and controlled also needs to be built in CanalCAD. The Fortran control logic
subroutine will be called inside the CanalCAD simulation file in order to model the control
effects.

The constants used in the final control algorithm (KP, KI, FC) must be determined through
simulation in CanalCAD and/or optimization in MatLab; the RF constants (from 0 to 1) are
achieved through the trial and error. These procedures are explained in the following
chapters.

Summary

The control algorithm of PIF or Reduction Factor and the associated heuristic adjustments to
be used for a pump or gate are rearranged and used to calculate the change in pump speed or
movement of the gate that is required to achieve a flow rate change for performing upstream
or downstream water level control or flow control.

For radial and other undershot gates, the head loss must be calculated for each control action:
The required gate opening change is:
\[
\Delta VGO = \frac{\Delta Q}{C_d \times (GN \times GW) \times \sqrt{2 \times g \times \Delta H}}
\]
For weirs in free flow condition \( Y_{ds} - U \geq (2/3) \times (Y_{us} - U) \), the algorithm uses the head between the upstream surface and the crest of the overshot gate; the required change in vertical gate height is:

\[
\Delta U = \frac{-\Delta Q}{3.09 \times GW \times GN \times 1.5 \times \sqrt{(Y_u / s - U)}}
\]

For weirs in submerged (flooded) flow condition \( Y_{ds} - U < (2/3) \times (Y_{us} - U) \), the algorithm uses the head between the upstream surface and the downstream water surface; the required change in vertical gate height is:

\[
\Delta U = \frac{-\Delta Q}{8.02 \times GW \times 1.5 \times \sqrt{Y_{u/s} - Y_{d/s}}}
\]

The pump speed change corresponding to the required flow change is different based on whether the TDH is a constant or changeable value.

In regards to recommendations for speeding up the delivery of water through a canal to a reservoir, a fixed flow controller at all gates over a certain period of time can be applied. A second option is a flow rate change as feedforward signal to all the gates and a ramp down of all set points over a certain period of time. One drawback of this method is that the water levels need to be steered back to the original setpoint after this period.

Other control methods that include control of the water level of the reservoir itself are worthwhile investigating. These methods will be more accurate, but require physical modifications to the system, like communication lines over longer distances.
CHAPTER 2. CANALCAD SIMULATION MODEL SOFTWARE

General Description

To run simulations of canal automation ITRC presently uses CanalCAD, which utilizes simulation technology based on algorithms developed by Cunge, Preissmann, Chevereau, Holly, and others at SOGREAH of France. CanalCAD has a user interface that was developed through a combined effort by ITRC, Imperial Irrigation District, and the Iowa Hydraulic Research Institute in the early 1990's. Since then, the program has been continually refined by ITRC, leading to Version 205 finalized in June 2001.

Due to these improvements, CanalCAD can now select any location within a pool as the "target" control location. This means that in the control algorithm subroutine, CanalCAD can extract water levels or flow rates from any designated point or even the head across a gate. (In the past, we were limited to 5 specific locations within any pool.) These designated points are assigned names in CanalCAD such as MTx1, MTx2, MTx3, MTx4, MTx5, etc.

Actual vs. Modeled Results

For the most part, control effects simulated in CanalCAD are very similar to actual effects seen in the field. In some cases, it has even been possible to duplicate unusual conditions in CanalCAD, such as harmonic gate movements or pump speed changes resulting from very low flows. An example of this is Patterson Canal. Patterson Canal uses pump control based on an intermediate point determined using the Bival method. In the field, it was found that the speed of Patterson’s Pump 1 reacted harmonically at very low flow rates and, as a result, was unable to provide effective control at these very low flow rates. ITCR was able to duplicate the precise action of Pump 1 in CanalCAD, which can be seen in the upper part of Figure 2-1. Based on the duplicated simulation, ITRC could try several methods to manually re-tune the FC (filter) parameter of the control algorithm, and finally selected one that provides optimal control. The improved control is show in the lower graph in Figure 2-1.

Patterson Irrigation District notified ITRC that the new control effects are very good, even at very low flows, and the harmonics problem with Pump 1 has been eliminated.
**Discrepancies with CanalCAD Results**

Occasionally the water levels predicted in CanalCAD do not match the real water levels monitored in the field. These discrepancies have been noticed during extreme conditions, but in fact we do not know how extensive they are. We know that in most cases, the control that we have predicted with CanalCAD is what we have obtained.

It should be noted that CanalCAD has given satisfactory results, except for a small bug that has appeared on two occasions in the program, regarding the elevation reading. Instead of the actual elevation, CanalCAD gave the incorrect value “-1.00E+30” (shown in Figure 2-2). This did not appear very often or at all in the simulation; it only showed up in the initialization reading from the 1st second. This bug does not have a large impact on ITRC’s simulations, but it has been brought to the attention of the developer.

![Duplicated harmonic flow of Pump 1 at very low flow.](image1)

![Improved flow effects of Pump 1 at the same very low flow.](image2)

**Figure 2-1. The duplicated harmonic flow and improved flow effects for Pump 1 at Patterson Canal**
Figure 2-2. The small bug regarding the MTx3 elevation in CanalCAD

Figure 2-3 shows an example of another discrepancy that we cannot yet explain. It is for a single pool that has downstream control at Sutter Mutual Water Company. The water level is controlled at about 82% of the distance down the pool, and a Variable Frequency Drive (VFD) pump controls the inflow to the pool. The commissioning of the installation required that a flow rate change be made at the far downstream end of the pool. Figure 2-3 shows that the actual control is better than what was predicted in CanalCAD.

Figure 2-3. Simulated and actual water level changes at Portuguese Bend Canal, Sutter Mutual Water Company
Overall, however, the control strategy selected for a canal based on the control predicted by CanalCAD in most cases results in satisfactory control in the actual field.
CHAPTER 3. CHARACTERIZING THE POOLS

The shape and size of a pool have a great effect on how easy it is to maintain the target water level at the target point and achieve good control. (Note: in this context, “pool” refers to the section of canal between two control structures. Seen in this way, a canal is composed of a series of connected “pools”.) The progress that has been made by ITRC as of today is that all the pools can be characterized as an entire unit (the canal) instead of each individual pool being characterized separately, as ITRC had to do before. The characterization examines how the pool’s size and shape will affect the control, which includes how much the water level will respond (increase) due to a change (increase) in flow ($A_s$), how long it takes a flow change to affect the water level at the target point ($\tau$), and how much the water level will be affected by waves in the water (resonance). These characteristics will be used to optimize the control constants in MatLab.

Surface Area ($A_s$) and Delay Time ($\tau$)

Introduction to $A_s$

When the flow into a pool is changed but the outflow remains constant, the water level in the pool will also change. If the inflow is increased, the water level will rise, and if the inflow is decreased, the water level will drop. The opposite would be true if the inflow was constant and the outflow was changed. If the outflow were increased, the water level would drop, and if the outflow were decreased, the water level would rise. The rate at which these changes occur would on the size and shape of the pool (i.e., a small pool would fill faster than a large pool) as well as the difference between the incoming and outgoing flow.

Even more specifically, this rate is dependent upon the current surface area of the water: to increase the water level by 1 foot in a pool that has a current water surface area of 5 ft$^2$ would requires at least $5 \times 5 \times 1 = 25$ ft$^3$ of water, but to increase the water level by 1 foot in a pool that has a current water surface area of 10 ft$^2$ would require at least $10 \times 10 \times 1 = 100$ ft$^3$ of water. If the flow into both pools were increased by 5 CFS it would take 5 seconds to raise the water level in the first pool and 20 seconds (twice as long and twice as wide) in the second pool. In other words, the rate of water level increase will be faster in a pool with a smaller water surface area than in a pool with a large water surface area.

Of course, a pool’s water surface area will be different at different water levels, depending on the shape of the pool. However, the ultimate goal of automated canal control is to maintain a certain target water level (corresponding to a specific flow rate at one end of the pool), regardless of any changes in flow anywhere else in the pool. If the water level does change due to a change somewhere in the pool, the goal is to initiate a control action that will compensate for this change and bring the water level back to target level and keep it there.
Two things are important to emphasize:

1) The rate at which the water level changes due to a change in inflow at the head point of the pool (downstream control) or in outflow at the end point of the pool (upstream control) is dependent on the pool’s current water surface area, and

2) The pool’s current water surface area depends on the current water level (how full the pool is). This means that the speed at which a target water level can be achieved depends on water level and its corresponding pool’s water surface area.

Usually, we already know the target water levels or the target water level range for various flows from the district. We do not, however, know their corresponding surface areas, so we don’t know how quickly the target levels can be achieved from the current high or low water level.

Normally we only use one target water level for both maximum and minimum flow rates when we determine the water surface area “As,” though we know a district might change the target depth at different flow rates.

Figure 3-1 below illustrates target water levels for maximum and minimum flows that a district normally would use in a typical canal and the target water level that we normally use when we determine the “As”.

In Figure 3-1, it is apparent that due to a typical canal’s sloping sides and the different target depth settings at different flow rates, the surface area at low flows is typically less than the surface area at high flows. The surface area at normal flows will be somewhere in between. However, from a static point, if we use the same target water level for both maximum and minimum flow rate, the water surface area is the same for both maximum and minimum flow rates if the pool bottom is level. Also, we can manually calculate the surface area, which at least gives an idea about an appropriate range of As values. This is helpful when the As needs adjusting, as described in Chapter 4.

On dynamic water, the control is more concentrated. Therefore, when the “As” (surface area) is determined, it must be determined from a dynamic perspective. Thus, for the green dashed line in Figure 3-1, at this target level, if we introduce a 25 CFS flow rate change for both maximum 500 CFS and minimum 50 CFS flow conditions for the same time period, such as 200 minutes, then at the end of this time period, how much would the water surface area be for both maximum and minimum flow rates? Is the “As” at maximum flow necessarily
higher than that at minimum flow? The answer is no, because the amount that the introduced
25 CFS flow change impacts the whole pool, and specifically impacts the target point water
level, is influenced by many factors like the bank roughness, the consistency of the pool
cross-section, if there is a siphon or culvert between the target point, control structure, pool
bottom slope, etc. For this reason, it is important to determine the water surface area ($A_s$) in
each pool at maximum and minimum flow conditions.

Before explaining this process in more detail, it is first important to note how the change in
water level at the target point is affected by its location in the pool.

**Introduction to Tau**

Depending on where the target point is located in the pool, it may take awhile for a change in
flow to have any effect on the water level at the target point. If the target point is far away
from where the flow change was initiated, there will be a delay between when the flow
change occurs and when any effects are seen at the target point. However, if the flow change
is initiated right next to the target point, the effects will be immediate.

Say a pool’s water level has suddenly dropped and a control action is initiated to raise the
water level back up again. How long it will take the water level at the target point to increase
depends on how close the target point is to the control structure. The delay time between
when the flow change is initiated at the controlled structure and when the water level at the
target point starts to respond is called $\tau_d$, Tau.

Since Tau (units in seconds) is dependent on the location of the target point relative to the
control structure:

1. For **upstream control**, where the target point is located immediately upstream of the
   control structure at the end of the pool (reaction is immediate), $\tau_d = 0$ or a low value such
   as 1-5 seconds.
2. For **immediate downstream control**, where the target point is located immediately
   downstream of the control structure at the head of the pool (reaction is immediate),
   $\tau_d = 0$ or a low value such as 1-5 seconds.
3. For **intermediate and far-end downstream control**, the target point is far away (at an
   intermediate or far-end location in the pool) from the control structure at the head of the
   pool. This means there is a delay between the initiation of a control action at the control
   structure and the reaction of the water level at the target point, so $\tau_d$ must be calculated;
   typically it is a larger number such as 5-20 minutes.

Theoretically, $\tau_d$ for upstream and immediate downstream control is always zero or a low
value for any flow condition. If, in reality, there is a delay between the moment when the
control action is initiated and when it actually takes effect, this extra delay is expected to be
only 1-10 seconds (at any flow rate). For this reason, $\tau_d$ is always given as zero or 1-10
seconds in MatLab for upstream or immediate downstream control. However, when
intermediate or far-end downstream control is used, Tau must be calculated for each pool.
Tau must also be calculated for each pool at minimum and maximum flow conditions.
Calculating $A_s$ and Tau

As mentioned earlier, the water surface area when the water level is at target value ($A_s$) reflects how hard it is to maintain that water level (or, how quickly the water level changes).

The general equation for $A_s$ is as

$$A_s = \frac{\Delta Q \times \Delta t}{\Delta H}$$  \hspace{1cm} (3-1)

where

$\Delta Q =$ the introduced flow rate change, normally a constant of about 5% of the max inflow at time of “$t_0$”. The level at this moment is $H_t$.

$\Delta H = H_t - H_{td}$, or, the change in raised water level due to the flow rate change between time “$td$” (the moment the level at the target location begins to respond to the flow rate change) and time “$t$” (usually ½ to 1 hour later, or longer).

To solve for $A_s$, a CanalCAD simulation will be run to simulate the flow conditions in equation 3-1. The whole canal model needs to be run for each individual pool separately for maximum and minimum flow rates in order to observe the rate of water level increasing and the delay time in each pool.

The pool will be initially maintained at (target water level – 0.01 to 0.05 feet). After a time, a flow rate change will be introduced at one end of the pool (while maintaining a constant flow rate at the other end) so that the pool begins to fill and the water level begins to rise. The flow rate change should mimic a control-initiated change, so the flow change will be initiated at the same end of the pool as the actual control structure.

For upstream water level control where the target location is at the end of the pool, normally the inflow at the head point of the pool is kept constant and the outflow rate at the end of the pool is dropped so that the water level at the target location can rise. For downstream water level control where the target location is at an immediately or intermediately downstream point of the pool, normally the outflow rate at the end point of the pool is kept constant and inflow at the head point is raised by a certain amount so that the water level at the downstream target location can rise.

More explicitly:

a. For upstream control, the inflow will equal the desired flow condition (high or low) and remain constant throughout the simulation. The initial outflow will equal the inflow (minus any turnout flows) and then be decreased at the time of $t_0$ (e.g. 40 minutes or 1 hour) by 5% of the max flow. This is achieved by keeping the gates at the pool head in place and dropping the undershot gates (or raising the overshot gate) at the pool end by a certain amount. This amount can be calculated based on Equation 1-3 (or Equation 1-8).

b. For downstream flow, the inflow will change but the outflow will remain constant. The inflow will initially equal a flow rate that is slightly more than the sum of the
flow rates of all the turnouts, and will increase at the time of \( t_0 \) (e.g. 40 minutes or 1 hour) by 5% of the max flow. The outflow will always equal the initial flow (the sum of all the downstream turnouts). This can be achieved by keeping the gate at the pool end in place and raising the undershot gates (or lowering the over shot gate) by a certain amount. This amount can be calculated based on Equation 1-3 (or Equation 1-8).

This simulation can utilize the pool’s actual control structure, which can be built into the CanalCAD model. All flow into the pool at the head and all outflows as well as the needed 5% flow rate change can be obtained by moving the gate inside the Control Fortran file.

After this simulation for all pools has been run in CanalCAD, its water level data over time for each pool is graphed in an Excel worksheet. A trendline is used to determine the approximate slope \( \frac{\Delta H}{\Delta t} \) of the data line (after the change in flow), which indicates the rate at which the water level rises. The figure below presents an example of the water level increase produced at low flow rate conditions (115 CFS) with an introduced flow rate change of 5% the max flow (5% X 1100 CFS) in the CCID Upper Main Pool upstream of Redfern Check (immediate upstream control, initial water depth = 6.633 ft and target water depth = 6.68').

![Figure 3-2. Water level increase over time, resulting from a 55 CFS (5% of max flow) inflow increase and a constant inflow (immediate upstream control)](image-url)
From the trendline equation in Figure 3-2, it can be seen that the water level rises linearly at a rate of 0.0038 ft/min (the slope of the trendline). The $A_s$ which causes this rate of increase can be calculated as follows:

$$ A_s = \frac{\Delta \text{Flow}}{\text{Rate of Water Level increase}} \quad (\text{ft}^2) \quad (3-2) $$

$$ A_s = \frac{50 \, \text{ft}^3 / \text{s}}{0.0038 \, \text{ft}} = \frac{789473.7 \, \text{ft}^2}{60 \, \text{s}} = 73346.29 \, \text{m}^2 $$

It should be mentioned that the water level line (dark blue) looks like one single line, but in reality, it is a combination of two lines: one line is for the time of 0 to 40 minutes; the other is the line from 40 minutes to 240 minutes. The separated drawings for the trend line and its associated equation are based on the second timeline for 40 – 240 minutes, in which the time at 40 minutes is the moment when the flow rate change is made and level starts to go up.

The trendline equation could also be used to solve for $\tau$. First, there are two types of $\tau$ – one is the $\tau$ that can be read visually, which is 0 seconds in Figure 3-2 (when the flow rate change is made, the level immediately goes up, which is normally what we find for immediate upstream or downstream control). The other type of $\tau$ is calculated based on the trendline equation. It is just a matter of solving for a point along a line:

For the trendline equation of $y = 0.0038x + 6.4926$, at the time that the flow change was made ($T = 40$ minutes, $y = 0.0038*40+6.4926 = 6.6446$), it takes $\tau$ time for the level to reach the target of 6.68, so the $6.68 = 0.0038 \times (\tau + 40) + 6.4926$. So, $\tau = 9.32$ minutes.

The interpretation of the calculation of this $\tau$ is: for a level of 6.49' in a low flow condition of 150 CFS; if a 50 CFS flow increase is made, then it will take 9.32 minutes for the level to reach the target of 6.68'.

Note that it is not necessary to calculate this $\tau$ for immediate upstream or downstream control; the theoretical $\tau$ value for immediate downstream (or upstream) control is always assumed to be 0 or a low value such as 1-5 seconds. This calculated $\tau$ value is only used for intermediate or far-end downstream control.

$A_s$ and $\tau$ will be used to optimize the control constants in MatLab. For input into MatLab, convert:

$$ \text{Tau from minutes to seconds} $$

$$ \text{"delayt"} = \tau \times 60 $$

$$ A_s \text{ from sq. feet to sq. meters} $$

$$ \text{"A_s"} = A_s \times 0.092903 $$
Resonance

Ground Frequency ($\omega_r$) and its Resonance Peak ($R_p$)

Ground Frequency ($\omega_r$)

As described by Shuurmans et al., 1999, ground frequency, $\omega_r$, is the frequency of wave travel in a pool, primarily determined by the pool’s physical structure and properties.

If a wave is initiated at one boundary of the pool, when it arrives at another boundary (e.g., a control structure), part of the wave is reflected. As a result, if the pool is short and level enough, waves will travel up and down the pool several times before they damp out. If the wave travels a round trip and returns to the initiated point after a period of $T_r$, then $T_r$ can be approximated as:

$$T_r \approx \frac{L}{V + \frac{c}{c-V}}$$ (3-3)

Where

- $L =$ Length of the pool, meters
- $c = \sqrt{\frac{gA}{W}}$, the wave celerity at the control location (m/s)
  - in which, $A =$ cross sectional area, sq. m.
  - $W =$ top width of the pool, m
  - $g = 26.312$, gravitational constant, m/s$^2$

$$V = \frac{Q}{A}, \text{ where } Q = \text{the flow rate through the area, (m}^3/\text{s})$$
  - $A =$ cross sectional area, sq. m.

Approximately, the wave round trip time is: $T_r = \frac{2L}{c},$ (seconds).

The wave frequency corresponding to this travel time period is the ground frequency, $\omega_r$:

$$\omega_r = \frac{2\pi}{T_r} \approx \frac{2\pi}{\frac{2L}{c}} = \frac{\pi c}{L} \text{ rad/sec}$$ (3-4)

From this equation it can be seen that the major factor influencing a pool’s ground frequency is the wave celerity (since, for a given pool, the total length is a known number). The celerity is essentially a calculation of the wave’s inherent speed – how fast a wave, uninfluenced by water flow currents or mass, moves. This speed is powered by gravity according to the depth of the water. The water depth, in turn, is directly related to the cross-sectional area of the pool. So, if the cross-sectional area varies over the length of the pool, a separate celerity needs to be calculated for each different section.
Normally, a pool has several different cross-sections along its length. These cross-sections are usually either trapezoidal or rectangular, though water may also travel through a section of siphon with a circular or rectangle shape. Essentially, the celerity is calculated for each different cross section and then used to determine the average celerity of the pool.

If the celerity is a measurement of the wave’s actual average speed through the pool, the ground resonance frequency, $\omega_r$, is a measurement of how long it takes the wave to travel one period of its length. Therefore, the ground frequency of the pool is equal to Pi divided by the time it took the wave to travel the whole pool, or:

$$\omega_r = \frac{\pi \cdot \text{Celerity}}{\text{Total length of the pool}}$$

(same as 3-4)

Within a pool, there are actually an infinite number of resonance frequencies, each a multiple of the ground resonance frequency $\omega_r$. In addition, the calculated ground resonance frequency $\omega_r$ is really just an approximation since the estimate of “c” is inaccurate because the top width and other dimensions of a pool change with distance. Therefore, the calculated $\omega_r$ is only roughly equivalent to the actual $\omega_r$.

For this reason, $\omega_r$ is no longer used as an input inside the MatLab optimization routines to get initial control constants. Rather, $\omega_r$ is simply used to gain an idea about the ground frequency of the wave that is determined by the pool’s physical dimensions and components. If needed, $\omega_r$ can be manually calculated for comparison with the critical frequency ($\omega_{cr}$) input for MatLab optimization routines. This will be explained in further detail in Chapter 4.

**Ground Resonance Peak ($R_p$)**

If $\omega_r$ is the ground frequency of a wave initiated at one boundary of a pool, $R_p$ is the maximum amplitude that such a wave would reach at the target location. However, due to the inaccuracies of the calculated $\omega_r$ mentioned above, the $R_p$ that was previously determined through a simulated process of trial and error using CanalCAD and a template file created by ITRC is no longer used in MatLab optimizations.

**Effects of the Physical Properties of the Pool on Resonance**

Resonance can have a large impact on the control stabilization within a pool. Resonance is more likely to occur when the whole pool is affected by the backwater curve.

Figure 3-3 shows the effect of changes in the canal properties on the resonance peak ($R_p$) and resonance frequency ($\omega_r$). The figure can be interpreted as an analysis of the influence of the properties on the resonance effects. Arrows indicate an increase or decrease in the parameter value of the particular property.
The farther away from the origin, the larger the effect. From Figure 3-3, it can be concluded that the friction parameter (also known as roughness or Manning’s) has the largest impact on the resonance peak. (Therefore, it especially important that this parameter is set correctly when making the CanalCAD model.)

Since, according to Shuurmans et al., the proportional parameter, KP, of a PI-controller is limited by \( \frac{1}{2 \cdot R_p} \), the \( R_p \) will greatly influence this control parameter, a key factor causing instability. For a pool, a large \( R_p \) will result in a smaller KP and looser control. A small \( R_p \) will result in a larger KP and tighter control. So, high resonance peak values will limit the “tightness” of the control. In Figure 3-3, the top of the figure can be considered to be the zone with the most problems and the loosest control, while the bottom can be seen as the zone with the least problems and the tightest control. However, an excessively tight control is susceptible to causing instability. The key is to find the critical resonance of the wave that is caused as part of the control action and to which the pool will most dramatically respond.
Critical Frequency ($\omega_{cr}$) and its Critical Resonance Peak ($R_{cr}$)

The ground resonance frequency, $\omega_{r}$, and its resonance peak, $R_{p}$, consider the resonance effects caused by a wave traveling up and down the pool. This is important when the target point is located somewhere in the middle of the pool (intermediate or far-end downstream control). However, when an immediate upstream or immediate downstream control is used in which the target point is right next to the control structure, the movement of the control structure also greatly impacts resonance. In these control situations, the critical resonance peak (instead of the ground resonance peak) must be used to describe the resonance of the pool. $R_{cr}$ accounts for the impact of control actions (gate movement, etc.) on the resonance at the target point.

Critical Resonance Frequency ($\omega_{cr}$)

The critical frequency and resonance ($\omega_{cr}$ and $R_{cr}$) are the parameters that describe a peak in the pool’s sensitivity to certain waves. The critical frequency $\omega_{cr}$ is computed with the following formula:

$$\omega_{cr} = \frac{\pi}{\max(\tau_{d,low}, \tau_{d,high}) + T_{c}} \text{ (rad/s)} \quad (3-5)$$

where:

- $T_{c}$: Control time step (s), normally 60 seconds
- $\tau_{d,low}$: Delay time at low flow (s)
- $\tau_{d,high}$: Delay time at high flow (s)

For upstream and immediate downstream control, the theoretical values for $\tau_{d,low}$ and $\tau_{d,high}$ are zero. If it is known that a delay exists between the moment that the control action is initiated and the time when the level at the target location is impacted, this extra delay should be figured into the optimization program. If no information is available it is still advisable, to stay on the safe side, to always add a delay time of 10 seconds.

Critical resonance frequency is determined by the time between two continuous gate or pump control actions and the time it takes for the water level at the target point to react after the control action is given. In CanalCAD, a control time step is used inside the control algorithm. In the field, the control time step is the time at which a control action is implemented. ITRC typically uses 1 or 2-minute control time steps for level control, and 3 to 10-min control time steps have been used for Replogle flumes and/or ADFM (Acoustic Data Flow Meter) flow control.
Critical Resonance ($R_{cr}$) and $K_{P_{cr}}$

As stated earlier, $R_{p}$ and the proportional parameter $K_{P}$ are inversely proportional, which can be expressed as:

$$KP \propto \frac{1}{|R_{p}|}$$  \hspace{1cm} (3-6)

When a PIF (or PI) control logic is used, $K_{P}$ is the key factor causing instability. So, if Proportional-only control is used to run a simulation and the $K_{P}$ is gradually increased, at a certain point control will be overshot and instability will result. This $K_{P}$ value is regarded as the critical $K_{P}$, or $K_{P_{cr}}$. Just as $K_{P}$ corresponds to $R_{p}$, $K_{P_{cr}}$ corresponds to $R_{cr}$. So, if we determine the $K_{P_{cr}}$ at which instability occurs, we can use it to calculate the $R_{cr}$.

When determining the $K_{P_{cr}}$, the simulated flows should be kept constant at realistically low flows (10% of max flow). $K_{I}$ (integral constant) and $F_{C}$ (filter constant) should be set to zero, so that $K_{P}$ will be the only factor influencing the control effects. $K_{P}$ should be given a low initial value, such as 20 (or –20 if immediate downstream control is used). Then an unsteady state simulation, in which the target level is changed by ±0.01’ every hour or half hour (ideally it should be more than 3-4 times the delay time) instead of changing the gate height or flow rate schedule, should be run and $K_{P}$ increased (or decreased, for immediate downstream control) by the same interval at a set increment to zoom in on the $K_{P_{cr}}$ that makes the pool unstable.

The disturbance of the set point is introduced at the same time that the $|K_{P_{cr}}|$ value is increased (e.g. every hour or half hour). The following formula can be used to create the disturbance:

$$sp = sp_{0} + 0.01 \times \cos \left( \frac{\pi \times t}{T_{inc}} \right) \text{ (ft)}$$  \hspace{1cm} (3-7)

where:

- $sp$: Set point to be used for the coming $T_{inc}$ seconds (ft)
- $sp_{0}$: Original set point (ft)
- $t$: Time (s)
- $T_{inc}$: Time interval after which the $K_{P}$ value is increased, e.g. 1800 or 3600 (s)

The result for upstream control for the Redfern check structure of CCID Upper Main Canal is shown below. Note that the initial $K_{P}$ value was 40, the unacceptable gate oscillations starts at 11 hours, and the $K_{P}$ was increased in increments of 2 every half hour, so the $K_{P_{cr}}$ at which the gate starts the unacceptable oscillation is $K_{P_{cr}} = 40+2*(11*2-1) = 82$. The example in Figure 3-4 is just for the pool upstream of Redfern check, which is run together with all other check structures in one complete run simulation.
Figure 3-4. KP<sub>c</sub> beginning with unacceptable gate oscillations

Usually, when the turbulence of the set point is initiated, the control action resulting from the stepped increasing |KP<sub>c</sub>| is also initiated to compensate for the changed target level. However, when the KP<sub>c</sub> has reached a certain point, the gate starts to go up and down forming the instability and the pool resonance is enhanced to the point that the water level fluctuations from the target don’t decrease, but actually increase over time. This can be seen in the figure above.

Once an accurate value for KP<sub>c</sub> has been obtained, the following formula is used to determine R<sub>c</sub>:

\[ R_{cr} = \frac{1}{KP_{cr}} \]

So, if the critical proportional gain, KP<sub>cr</sub>, for the upstream control shown in Figure 3-4 is 82, then the critical resonance peak is:

\[ R_{cr} = \frac{1}{82} = 0.0122 \]

Note that the KP<sub>cr</sub> found by the process outlined above and used in this equation is not the same as the KP that will eventually be determined by MatLab, or used in the final CanalCAD simulation. MatLab redefines and optimizes the value in conjunction with a filter FC and an integral KI, so the final result is normally a higher KP together with KI and FC that can safely be used.

The resonance peaks are calculated for all pools based on the MINIMUM flow rate. This is because at low flow rates the pool is most sensitive to resonance waves. It is important that the critical resonance peak be determined for the worst-case scenario. This means considering more than just flow rate. For example, when flashboards or long crested weirs are lowered, the damping function of waves is to be expected. So, in this case, the worst situation would be to fully raise the flashboards or long crested weir and let all the flow go through the radial undershot gate or overshot gate.
Using a Small Computational Time Step

When running simulations in CanalCAD to determine $R_p$ and $K_{P_r}/R_{cr}$, a computational time step that provides the best possible simulations of wave actions, flows, etc. should be used. This will usually be a value that ranges from 1 second to 10 seconds, depending on the pool characteristics. (When determining the “$A_s$” and “$\tau$”, since there is no control action, control structure, or resonance wave involved, and since we only introduce flow change and care about the water level rising speed with time, ITRC always uses a 1-min computational time step to determine the “$A_s$” and “$\tau$”. This never gives problems.)

There are two considerations in selecting the proper computational time step for CanalCAD when involving control:

a. For very large canal systems, such as the DMC, using a 1-second time step to simulate an event of longer than about 18 hours overloads the memory capacity of the computer. This is an important consideration, but it can be overcome by simply simulating events of less duration.

b. For short pools, if the time step is greater than 1 second, CanalCAD can easily “miss” the resonance wave shape. That is, if the time step is 1 minute, resonance waves may not even show up in the CanalCAD output. This is the most important item, in terms of impact on the accuracy of our recommended control algorithm.

As one changes the computational time step from 1 second to 2 seconds, and goes upward in increments, it can be seen that for a given pool, at some computational time steps the results begin to look different. That is, there is some minimum acceptable time step that will give “true” results for any pool. If one uses a larger time step, the results will be erroneous.

The minimum acceptable computational time step, $T_{comp}$, to be used in CanalCAD can be estimated from the critical frequency. For each computational time step, one point on the critical frequency sine wave is calculated, so $T_{comp}$ must be small enough to calculate enough points to effectively reflect the whole shape of the sine wave. For a sine wave to be computed accurately, approximately 50 points (absolute minimum would be 20 points) need to be calculated during its period of frequency. Therefore, if we take 30 points for a one-period sine wave, the computational time step can be determined with:

$$T_{comp} \leq \frac{2 \cdot \pi}{30 \cdot \omega_{cr}} \quad \text{and} \quad \omega_{cr} = \frac{\pi}{\max(\tau_{d,\text{low}}, \tau_{d,\text{high}}) + T_c}$$  \hspace{1cm} (3-9)

As stated earlier, for upstream or immediate downstream control, $\max(\tau_{d,\text{low}}, \tau_{d,\text{high}})$ normally equals zero. So, for an upstream or immediate downstream control simulation with a control time step of 1 minute, the minimum acceptable computational time step would be:

$$T_{comp} \leq \frac{2 \cdot \pi}{30} \cdot \frac{\max(\tau_{d,\text{low}}, \tau_{d,\text{high}}) + T_c}{\pi} = \frac{2 \cdot 60}{30} = 4 \text{ (sec)}$$

From this, we infer that the computational time step used in the CanalCAD simulations should be kept as small as possible, preferably 1-4 seconds; even 5 seconds would be
acceptable. However, if using a 1-second computational time step results in data loss because of limited computer memory, the time step should be gradually increased to 2 to 5 seconds, still keeping it as small as possible. The use of a large computational time step is prone to “shorten” the actual disturbance shapes and calculates a smaller $R_p$ or $R_{cr}$.

Figure 3-5 illustrates the difference in resonance peak determined by using a 1-sec, 5-sec, and 1-minute computational time step. It can be seen that the small computational time steps of 1 second and 5 seconds give similar results, but a large time step (1 minute) gives very different simulation results – which would cause incorrect conclusions. Figures 3-6 and 3-7 show the different control effects resulting from 1-sec and 1-min computational time steps at the Umatilla Stanfield Branch Furnish Canal. These two figures use PI logic to control a combination of over-shot and under-shot gates for immediate downstream control with a control time step of 1 minute. (These figures are from 2002; later that year, ITRC improved the control logic from PI to PIF and the new logic was implemented.)

Using a 1-min computational time step in our first simulations (Figure 3-6) caused us to miss the gate oscillation and pool resonance problems shown in Figure 3-7, which in the field caused the gate panel heaters to overheat and prevented the gates from moving.

![Figure 3-5. The difference in the resonance peak determined by using 1-sec, 5-sec, 1-min computational time steps](image)

Irrigation Training and Research Center
Figure 3-6. Downstream control simulation results from the initial tuning of PI algorithms on the USBF Canal using a 1-minute simulation time step and a 1-minute control time step.

Figure 3-7. Downstream control simulation results on the USBF Canal using a 1-second simulation time step and a 1-minute control time step, still using the PI algorithm.
Summary Points

- Two values for $A_s$ and $\text{Tau}$ need to be calculated for each pool at maximum and minimum flows.

- Calculate $R_p$ for each pool at minimum flow for immediate upstream or immediate downstream, intermediate downstream or far end downstream control.

- Use a 2-5 second (or as small as possible) computational time step when determining $R_c$. Use a 1-minute computational time step when determining “$A_s$” and “$\text{Tau}$.”
CHAPTER 4. OPTIMIZING THE CONTROL LOGIC PARAMETERS

Background

ITRC continues to try to develop or locate a procedure that tunes the PI or PIF algorithms satisfactorily and quickly. Historically, ITRC systematically ran simulations starting with one set of KP and KI constants, and varied the KP and KI constants step-by-step in an attempt to bracket the best values. This was a trial-and-error technique that was complicated by the need to use different KP and KI constants for each pool. Because of pool and gate interactions, the tuning of all controllers must be done simultaneously. This procedure typically required a minimum of one month of systematic tuning using CanalCAD, and often took longer.

For the past five years, ITRC has been working with Jan Schuurmans and Peter-Jules van Overloop from the Netherlands to develop a better and quicker systematic tuning procedure. The procedure is to first determine resonance and wedge storage characteristics for each individual pool using CanalCAD (refer to Chapter 3), and then to use special MatLab® routines based on the hydraulic characteristics of surface area, resonance peak and delay time to simultaneously optimize the unique PI or PIF constants for each gate or pump controller. These MatLab routines have been continuously updated and are now getting to a start-of-art point that can directly optimize a set of parameters without the need to finely tune them.

How MatLab Optimization Works

The MatLab optimization routine developed by Schuurmans and Overloop gives the optimum KP, KI, and FC values. The whole program is a combination of more than ten subroutines, which each perform a different function. All subroutine file names end with the extension “.m”. There is one template “.m” file that the user uses to change the program inputs. These inputs include the pool characteristics determined using CanalCAD, described in Chapter 3. To optimize the control parameters, the user opens the template “.m” file, changes the input values, and saves the file under a new name in the default folder of “c:\MatLab\Work”. This is the only MatLab file that will be changed by the user. Then, to initiate the optimization, in the MatLab Workspace the user types in the name of the saved file after the command “>>”.

What the user does is fairly simple. The internal optimization process, however, is very complicated. ITRC has tried to fully understand each step in the optimization process, but some aspects are still unclear. This section is an attempt to explain ITRC’s understanding of the internal optimization process at this time.

In the input file, when “control_logic = ’allowal’” is selected, the tuning code will automatically determine, for each pool, whether PI control or PIF control should be used. To
make this determination, it will first compute initial stabilizing times for each pool for both PI and PIF. Then, it checks if the stabilization time \( T_f = \frac{1}{\omega_{cr}} \) would actually be useful (if \( T_f > \frac{1}{\omega_{cr}} \), where \( \omega_{cr} \) (critical frequency) is internally calculated by MatLab as \( \pi/(\text{CtrlTime step} + \text{max}(\text{Tau})) \), where \( \text{max}(\text{Td}) \) is the pool’s highest delay time matrix value (either high or low flow Tau)).

If this is true (meaning that the pools are dominated by resonance waves rather than by the delay time), then the pools should be controlled with PIF \( (F_c > 0) \) (PIF control will be selected). Examples where this might occur are in short pools, flat pools, or pools with a target location immediately upstream or downstream, which operates under a low flow. With PIF control, the filter gain \( (F_c) \) and the proportional gain \( (K_P) \) will be kept the same as that initially calculated by the equation listed below during optimization, so that their values will not be increased during the process (which could destabilize the canal pool).

In general, if either the delay time or the control time step for a pool is “long” such as 3-15 minutes (for example, in a steep pool that has its target point at the remote downstream end of the pool), these pools should be controlled with PI-control \( (F_c = 0) \). If the delay time and control time step are “short” such as 1-3 minutes, the program is likely to select PIF.

After the MatLab optimization routine is run, an estimate of the optimum \( KP, K_I, \) and \( FC \) values is given. For “resonance dominated” pools (upstream and immediate downstream control), the MatLab optimization routine typically works in the following way:

\[
\frac{-T_c}{\min(A_{s,low}, A_{s,high}) \cdot R_{cr} \cdot \left( \max(\tau_{d,low}, \tau_{d,high}) + T_c \right) / \pi}
\]

- \( FC \) is set to a fixed value of \( e^{(4-1)} \)

- \( KP \) is limited to a fixed maximum of

\[
\frac{1}{2 \cdot \text{mean}(A_{s,low}, A_{s,high})} \min(A_{s,low}, A_{s,high}) \cdot R_{cr} \cdot \left( \max(\tau_{d,low}, \tau_{d,high}) + T_c \right) / \pi
\]

Where:

- \( T_c \) Control time step (s)
- \( A_{s,low} \) Surface area at low flow (m²)
- \( A_{s,high} \) Surface area at high flow (m²)
- \( \tau_{d,low} \) Delay time at low flow (s)
- \( \tau_{d,high} \) Delay time at high flow (s)
- \( R_{cr} \) Critical resonance peak input

- \( K_I \) is optimized in the iterative steps by minimizing the water level deviations and control actions for both low and high flow conditions.

For “delay dominated” pools (intermediate or far-end downstream control) the MatLab optimization routine works in the following way:
FC = 0

KP and KI are both optimized in the iterative steps by minimizing the water level deviations and control actions over both low and high flow conditions.

The MatLab tuning code uses the control equations to optimize the control parameters – just KI when PIF is used for immediate downstream and upstream control, and KP and KI when PI is used.

The PIF control equation can be represented by

\[
\begin{align*}
    e_f(k) &= (f_c)e_f(k-1) + (1-f_c)e(k-1) \\
    u(k) &= u(k-1) + k_p(e_f(k) - e_f(k-1)) + k_i e_f(k-1)
\end{align*}
\]

Where:
- \(e\) = the control error (m)
- \(e_f\) = the filtered error (m)
- \(u\) = the control flow input (m³/s),
- \(f_c\) = the filter constant
- \(k_p\) = the proportional gain
- \(k_i\) = the integration time constant

The PI control equation is similar to that for PIF, except that the filter is left out:

\[
\begin{align*}
    u(k) &= u(k-1) + k_p(e(k) - e(k-1)) + k_i e(k-1)
\end{align*}
\]

The tuning is based on minimizing the objective function, \(f\), according to:

\[
\min_{x} f, \text{ subject to } Dx = c \text{, where } x \text{ is a vector containing } k_p, k_i \text{ and } f_c \text{ values.}
\]

The objective function, \(f\), is given as:

\[
f = \sum_{k=0}^{\infty} e_1^2(k) + e_2^2(k) + \ldots + e_N^2(k) + r_1 \Delta u_1^2(k) + r_2 \Delta u_2^2(k) + \ldots + r_N \Delta u_N^2(k)
\]

Where:
- \(\Delta u(k) = u(k) - u(k-1)\)
- \(r_i = \text{input_weight}(1)\), i.e. the first element of the vector \(\text{input_weight}\) (defined in the input file).

For PIF control, when the KP and FC values are not optimized, the constraint \(Dx=c\) is used to fix \(k_p\) and \(f_c\). However, KI is always optimized. For PI control, FC is set as 0 for each pool, and KP and KI are both optimized.

The KP, KI, and FC values are then used within CanalCAD in a Fortran control subroutine to simulate the actual control of the canal. If needed, the fine-tuning of the \(k_p\), \(k_i\), and \(f_c\) values can be accomplished by running numerous simulations and observing the magnitude of gate movements and water level changes. However, it is strongly recommended that the pool characteristics that are put into the optimization be analyzed again and changed. For some reason the storage area might not be estimated correctly (possibly due to a structure limiting the flow through an area and, as a result, changing the available storage area) or
there might be some delay time not taken into account. Another possibility is that $R_c^r$ was not estimated at the actual worst-case (least damping) scenario.

Also, changing $R_c^r$ (for example making it higher if a pool is instable) rather than KP (making it lower manually) and re-running MatLab has the advantage that all KP, KI and FC are automatically changed accordingly. The effect that this change might have on the other values is also taken into account when the optimization is rerun.

**Decoupler Optimization inside the MatLab Routines**

The extension downstream controllers with decouplers should be implemented best as follows:

- In pool 1: $\Delta q_1(k) = (PIF \ or \ PI) \times e_1(k) + kd1 \times q_2(k-1)$
- In pool 2: $\Delta q_2(k) = (PIF \ or \ PI) \times e_2(k) + kd2 \times q_3(k-1)$
- In pool 3...for each consecutive pool

Where:

- $\Delta q_1$ = the change in controlled inflow to pool 1
- $e_1$ = the control error in pool 1 (= deviation of controlled water level)
- $kd1$ = the decoupler constant in pool 1
- $k$ = the sample time

Notice that the control system for pool 1 cannot obtain the $\Delta q_2$ change in flow at time $k$, since at time $k$ the $\Delta q_2$ is unlikely to be available yet. However, at time $k$, the previous value of $\Delta q_2(k-1)$ should be available. This is how it is implemented inside CanalCAD.

Figure 4-1 contains a sketch of the decoupler principle.
The decoupler constant $k_d$ is not necessarily best chosen at 1. When testing the decoupler at Corning Canal, selecting a $k_d$ as 1 resulted in a huge improvement over using an optimized PI or PIF without decouplers. However, there was still the effect of amplification from pool to pool. Figure 4-2 shows the response of the Corning Canal equipped with PI/PIF downstream controllers, but no decouplers. These controllers were optimised using the January 2004 version of the TUNE program. The turnout flow change at the downstream end of the Corning Canal was $1 \text{ m}^3/\text{s}$.
When decouplers are used that send the flow rate change of downstream feedback controllers to the upstream structures as a feedforward signal, the parameter Maxiter must be set to 0 (zero). In that case the control parameters KP, KI, and FC come out un-optimized. The decouplers should use a decoupler to gain a $K_d$ equal to 1. Figure 4-3 shows the response of the same PI/PID controllers with decoupler constants $kd=1$. The constants of the PI/PID controllers were computed using TUNE, but not optimised (by selecting Maxiter =0). The water level errors have been reduced considerably due to the decouplers, but the water levels in the first three pools show considerable oscillations due to disturbance amplification.

Figure 4-2. Canal controlled with optimized PI/PID controllers (no decouplers) using MatLab TUNE (January 2004 version).

Figure 4-3. Canal controlled with UNoptimized PI/PID controllers, extended with decouplers ($kd=1$)
Schuurmans and Overloop next investigated the effect of optimising not only the PI/PIF constants of kp and ki for each pool, but also the decoupler constant kd. The optimised constants are:

<table>
<thead>
<tr>
<th>Pool#</th>
<th>kp</th>
<th>ki</th>
<th>kd</th>
<th>fc</th>
<th>fcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-24.371</td>
<td>-0.57963</td>
<td>0.97235</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-18.3769</td>
<td>-0.093324</td>
<td>0.93845</td>
<td>0.90045</td>
<td>0.99825</td>
</tr>
<tr>
<td>3</td>
<td>-14.1291</td>
<td>-0.12546</td>
<td>0.83189</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-15.8812</td>
<td>-0.098709</td>
<td>0.79424</td>
<td>0.90391</td>
<td>0.99832</td>
</tr>
<tr>
<td>5</td>
<td>-11.1784</td>
<td>-0.13613</td>
<td>0.75508</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-10.563</td>
<td>-0.10028</td>
<td>0.85119</td>
<td>0.85711</td>
<td>0.99743</td>
</tr>
<tr>
<td>7</td>
<td>-8.6867</td>
<td>-0.10392</td>
<td>0.62844</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>-7.2014</td>
<td>-0.21833</td>
<td>0.97105</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>-9.1027</td>
<td>-0.32117</td>
<td>0.94994</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>-10.9</td>
<td>-0.2594</td>
<td>1.1248</td>
<td>0.74082</td>
<td>0.99501</td>
</tr>
<tr>
<td>11</td>
<td>-5.913</td>
<td>-0.068189</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4-4 shows the resulting response with the optimized controllers plus decouplers: clearly a considerable improvement over the unoptimized controllers of Figure 4-2 and 4-3.

![Figure 4-4. Canal controlled with optimized PI/PIF controllers, extended with decouplers](image-url)
Improving Unsatisfactory Results

To determine if the initial MatLab tuning results are satisfactory or not, their performance should be verified in CanalCAD. The optimized KP, KI, and FC constants should be input into CanalCAD as internal “regcon” variables for each pool. CanalCAD v205 Custom2 Mode provides 40 “regcon” variables. The values for these regcons can vary for each controlled structure in the canal model. After KP, KI, and FC have been input into CanalCAD (for each pool), they are called by the Fortran control code whenever a CanalCAD simulation is run. Normally, ITRC simulates a variable flow schedule for 1 or 2 days to see if the water level error and the controlled structure responses are satisfactory.

As the MatLab routines were updated in January 2004, its optimized PIF constants for CCID Upper Main Canal upstream control were tested with satisfactory results and stable control for both high and low flow conditions. In March 2005, the MatLab routines were updated to add decoupling, so that decoupling is used to send the downstream feedback controllers’ flow rate change to the upstream structures as a feedforward signal. This is a gradual and exciting progress, and the effects of the MatLab optimized constants for CCID’s other canals need to be verified.

Effects of Changing the $R_{cr}$ and $A_s$ Input Values

Simply changing the $R_{cr}$ and/or $A_s$ input values can improve unsatisfactory control results. The equations for FC and KP (equations 4-1 and 4-2) explain that $A_s$ and $R_p$ influence the values of KP and FC, except for the control time step $T_c$ of 60-120 seconds and Tau of 1-10 seconds for immediate upstream and downstream control (or longer for intermediate and far-end downstream control), which are normally unchanged for a specific control. Below is an explanation of the effect of changing the $R_{cr}$ and/or $A_s$ values.

Changing the $R_{cr}$ Input Value

| $R_{cr}$ down | $|KP|$ up | gate movement | Shortcoming | Cycling or too much gate movement |
|---------------|----------|---------------|-------------|----------------------------------|

| $KP_{cr}$ up | $R_{cr}$ down | $|KP|$ up | gate movement | Shortcoming | Cycling or too much gate movement |
|---------------|---------------|----------|---------------|-------------|----------------------------------|

Changing the “$A_s$” Input Value

| $A_s$ down | $|KP|$ up | gate movement | Shortcoming | Not enough gate movement and bigger water level error |
|----------------|----------|---------------|-------------|-----------------------------------------------|

Irrigation Training and Research Center
Though changing the $R_{cr}$ and/or $A_s$ can optimize out different KP, KI, and FC constants that will eventually result in more satisfactory control, we do not suggest changing the $R_{cr}$ and/or $A_s$ inputs without the support of canal physical characteristics. Each time a changed $R_{cr}$ and/or $A_s$ are input, they should be compared with the hand-calculated $A_s$ and $\omega_r$ values so that they are not too far apart.

Besides the methods that were illustrated above on how to change the MatLab input file for getting better constants, some other methods can also be made outside of MatLab for better control effects, as described below.

**Manually Changing FC to Improve Gate Movement**

Sometimes unstable gate or pump movements can be solved, or reduced, by manually changing the FC constants for each pool in CanalCAD. A manual FC adjustment, along with the initially optimized KP and KI values, can provide good control.

Figure 4-5 compares the control effects at CCIDLMC’s Checks 3 and 4 when the initially-optimized FC value is used and when an increased FC is used (initially-optimized values for KP and KI are used in both simulations).

![Figure 4-5. Control effect comparison for CCIDLMC Checks 3 & 4 using the original FC and an increased FC at low flow conditions](image)
The upper part of Figure 4-5 shows the control effects at Checks 3 and 4 when the initially optimized parameters were used. These parameters resulted in extremely excessive gate movement. In an attempt to reduce the gate movement, the FC for each pool was manually increased in CanalCAD. From the initial simulation results, a visual estimation of the amount of excessive gate movement was made for each pool, and the FC increased accordingly (i.e., each pool’s FC may have been increased by a different amount). The lower part of Figure 4-5 shows that there is much better control with the larger filters.

Sometimes increasing the initially optimized FC is enough to improve results satisfactorily. This was the case for Patterson Canal in Chapter 2 (CanalCAD, Figure 2-2) of this report. However, just increasing the FC was not enough to completely eliminate erratic gate movement. Figure 4-5 still shows signs of a harmonic gate. If this is the case, there might be something wrong with the $A_s$ and $\omega_{cr}$ inputs. These should be checked to see if there is any reason to readjust $A_s$, $\omega_{cr}$ and/or $\tau$, and to rerun the MatLab optimization routines.

### Resetting FE2 as Zero

FE2 should be reset as zero whenever the target depth changes more than an absolute value of 0.1”, whenever automatic control starts, or whenever an anti-hunt condition occurs. In the PIF control algorithm that is used for upstream or downstream control, the calculated change $Q$ considers the water level error that occurred at the previous control time step through a filter constant FC. The form is the same as (1-5):

$$
\Delta Q = (KP \times (FE1 - FE2)) + (KI \times FE1)
$$

(same as 1-5)

Where

- $\Delta Q =$ change in flow rate through the gate, in $\frac{m^3}{sec}$
- $KP =$ the proportional constant
- $KI =$ the integral constant
- $FE1 =$ the filtered error for the present time step
  
  $$
  FE1 = (FC \times FE2) + [(1 - FC) \times ENOW]
  $$

  where
  
  - $FC =$ the filter constant
  - $FE2 =$ the value of FE1 for the previous time step
  - $ENOW =$ present unfiltered error

In equation (1-5), the FE2 is indeed the filtered water level error at the previous control time step. It is this FE2 upon which the visible FE1 is calculated, with the aid of the filter constant FC and the present unfiltered error $ENOW$.

Previously, ITRC only suggested resetting the FE2 as 0 whenever the PLC power was given. However, for implementation, a district may want to change target water depth based on the real flow rate, or they may prefer to manually adjust a gate for some reason and then return to automatic control. Additionally, sometimes the anti-hunt condition occurs, which means that the water level error is bigger than a certain amount but the gates are moving in the wrong
direction. This is the result of an incorrectly recorded FE2 from the previous filtered error. This is because:

- When the target is changed, the correspondingly changed ENOW is not the result of the last control action, but the result of human intervention.

- When the control is turned from auto to manual, the PLC inside is still performing the calculation; ΔQ, FE1 and FE2 are still being calculated in each control time step, but the FE1 and FE2 values are now meaningless.

- When anti-hunt happens, the previous error will dominate the gate movement more than the present filtered error.

Therefore, ITRC now suggests that FE2 be reset to zero:

- When time is zero, or

- Whenever automatic control starts (after switching from manual to "auto" or after a power failure), or

- When the target depth changes by more than an absolute value of 0.1', or

- When anti-hunt occurs

Using a FE2 Scaler to Improve Upstream Control Effects

FE2 is the filtered water level error from the previous control action. Before, after time>0 and in the midst of any auto control executing, whether for immediate upstream or downstream control, ITRC always set FE2 = FE1. With this setting, the wave that has accumulated in each pool and in several adjacent pools can be damped significantly.

The idea of FE2Scaler is that if we can reduce the consideration or effect of a previous error on this control action, we can probably improve the control effects. With this thought, a scaler was introduce to the FE2 and gave a much improved control effect, which is presented in Figure 4-6.

Figure 4-6 is the comparison between control with and without an FE2 scaler for Gila Gravity “Y” check upstream control, which is a very large pool about 70000’ ft long. The upper left graph shows the entire 60-hour simulation without FE2scle, which means setting FE2=FE1, as previously set by ITRC. The upper right graph is an expansion of a 2-hour section (35.8-37.8) of the upper left graph without FE2scle. The lower left graph is another 60-hour simulation using a FE2scle of 0.9, which means setting FE2=0.9*FE1. The lower right graph is the 2-hour section (35.8-37.8) of the lower left graph without FE2scle.
Figure 4-6. The improved control effects of introducing an FE2 scaler for Gila Gravity “Y” check upstream control

From Figure 4-6 it can be seen that the water level reached zero much quicker using a FE2scle of 0.9, and the maximum water level error did not increase. Also, from the lower expansion graph, it can be seen that within the first 20 minutes of a big flow rate change, there were three small periods of downward gate movement where the water level error was greater than 0. After the first 20 minutes, the gates went smoothly up in relation to the above-zero water level error and then stopped fully.

At the same time, ITRC found that this FE2scle only improved upstream control, and that the FE2scle should be within the range of 0.1 to 0.9. For immediate downstream control, FE2scle gave no significant improvement. The FE2 Scaler and FC should also not be used for intermediate downstream or far-end downstream control where there are several delays or longer than 10 minutes of delay time between the initiation of a control action at a check structure and the time at which the far target location starts to react.

**Tuning Summary**

The pool characteristics determined through CanalCAD simulation are used in a MatLab routine to get the optimized control constants KP, KI and FC.
Once the pool characteristics are known ($A_s$ and $\text{Tau}$), 2 flow rates – minimum and maximum, and $K_{Pcr}$ ($R_{cr}$) are entered into a special numerical MatLab optimization program. This program also requires the input of other internal parameters for optimization, such as rho, term_tol, input_weight, and sample time (control time step).

For PIF control logic inside MatLab, the KP and FC parameters are calculated first, according to $A_s$, $R_{cr}$, $\text{Tau}$ (delay time), $T_c$ (control time step). Based on the initial KP and FC values, $KI$ is optimized. Once the KP, $KI$, and FC values have been optimized in MatLab, they are used within CanalCAD in the Fortran control subroutine to simulate the actual control of the canal.

The pool characteristics that are used inside the MatLab optimization routines must be checked against physical data (calculation of $A_s$ = surface of backwater and $\text{Tau}$ = delay).

Not only the resonance peak but also the resonance frequency need to be identified from the closed loop response test. Ideally, these data should also be checked against physical information, as with $A_s$ and $\text{Tau}$. However, it is much more difficult to invent simplified physically based formulas for these properties. For instance, the resonance may be due to badly dampened waves running up and down the canal, but they may also be caused by the sampling in the controllers (the sample time acts then as a delay).

Fine-tuning of the KP, $KI$ and FC values is normally accomplished by running numerous simulations based on different high and low flow scenarios and observing the magnitude of gate movements and water level changes.

The KP, $KI$ and FC values that are calculated in the MatLab routine by adjusting the $A_s$, $R_{cr}$ inputs currently can reach satisfactory control for low flow, high flow, and very low flow conditions. Depending on the initial constants generated in MatLab, the final control is normally tighter and smoother than the initial simulations because of the heuristic adjustments that ITRC designed.

In the end, a compromise must usually be made between water level control and gate or pump control. The KP and $KI$ combination that produces a quick stabilization with minimal overshoot should only be chosen if adjacent KP and $KI$ combinations also provide stable solutions. Because of our uncertainties regarding simulated versus true results, we never want to select tuning constants that are on the verge of causing instability.

If a small water level error is desired and quicker, larger gate or pump responses can be tolerated, the absolute value of KP should be increased by increasing the $A_s$ input and/or decreasing the $R_{cr}$ input (increasing $K_{Pcr}$) and re-optimizing for all control constants. On the other hand, if it is desirable to mitigate the gate or pump response, and a larger water level error can be tolerated, the absolute value of KP should be decreased by decreasing the $A_s$ input and/or increasing the $R_{cr}$ input (decreasing $K_{Pcr}$) and re-optimizing for all control constants.
Sometimes, when the gate reaches its opening or closing limit, the water level error may look overly large within the limit period; if this can not be ameliorated by adjusting the tuning constants, then the turnout scheduling might need to be changed.

Occasionally, the FC also needs to be manually changed in CanalCAD (after or without re-optimization) to eliminate unstable gate or pump movement. However, ITRC does not suggest changing FC without trying to change the $A_s$ and $R_{cr}$ inputs inside MatLab to re-optimized control constants.

Throughout the process of tuning, it is recommended that the minimum and maximum flow rates in the turnout schedule for each pool be **realistic** and kept **consistent** with those originally used to calculate $A_s$ in CanalCAD.
CHAPTER 5. RECOMMENDATIONS AND LIMITATIONS

Algorithm Choice: Reduction Factor Control with Replogle Flumes

When working on the flow control used for CRIT’s Check 56, which is based on the water head above a Replogle flume sill, ITRC found that Reduction Factor (RF) control, when used with a small Reduction Factor (RF), had the same effects as PI control. In the future, this knowledge will simplify similar flow control projects using Replogle flumes, Parshall flumes, Cipoletti weirs, electronic flow meters, etc. Much of the PI-related work, such as determining the best KP, KI, etc. parameters, can be omitted by using RF control with a small Reduction Factor; for example, from 0.2 to 0.4, with a 2 to 5 minute control time step, which will greatly decrease the wearing of the gate mechanic fittings.

Figures 5-1 through 5-3 present the control effects of using PI logic and using Reduction Factor logic, with RF = 1 and RF = 0.28, respectively. It can be seen that control with a small Reduction Factor RF of 0.28 in Figure 5-3 has the same effects as using the PI tuning in Figure 5-1. However, when Reduction Factor control with RF = 1.0 is used, so that ΔQ directly equals (QT - QNOW), the gate will experience overshoot cycling for awhile, but will eventually stabilize. This can be seen in Figure 5-2.
Tuning Algorithms for Automated Canal Control

Gate Opening or Water Depth (ft)

12 2400
10 2000
8 1600
6 1200
4 800
2 400
0 0

Depth just u/s of C56
Depth just u/s of flume
Gate Opening
Flume Flow

KP = -15.2 x 1.2 = -18.2
KI = -7.6 x 1.5 = -11.4
Ramp time for target water depth change = 0

Control TS = 1min
Computational TS = 3sec

Figure 5-1. Simulated effects of using PI control at CRIT Check 56

Flume Flow Ramp time for target water depth change = 0

Figure 5-2. Simulated effects of using Reduction Factor control with a RF=1 at CRIT Check 56
However, from the Reduction Factor equation of $DU = \text{Gate Position} \times \frac{RF \times Q\text{NOW}}{Q\text{NOW}}$, one can find that there are two limitations of using Reduction Factor for radial or undershot gates:

a. When the gate position is too low, such as if Gate Position = 0.05', it is susceptible to calculating a very small DU that cannot pick up the gate at all. The solution for this is: If the Gate Position is lower than 7% of the maximum gate opening, then the logic will always set the gate opening as 7%*Max Opening.

b. If the flow through the gate is too small, such as if QNOW = 0.05 CFS, then it is susceptible to calculating a very large DU that gives too much gate movement. The solution for this is: If the QNOW is less than 7% of the maximum flow rate, then the logic will always set the QNOW as 7%*Max flow rate.

The solutions offered in “a” and “b” above are not enough to improve the control. It is also necessary to have some heuristic adjustment for Reduction Factor control in order to improve the control effects. The improvement can be seen by comparing Figures 5-4 and 5-5 (improved).
Figure 5-4. The control effect of using Reduction Factor with the “a” and “b” adjustments

Figure 5-5. Improved control effect with “a” and “b” and heuristic adjustment as well
(comparable with Figure 5-4)
The improvements that are circled in green in Figure 5-5 were brought about with the added heuristics for gate movement. These can be categorized as follows:

1. *Flow rate control dead band* – If the |flow rate error| is less than the dead band (a user input value; 1 CFS is used in the simulation), then the gate should not be moved.

2. *Gate accumulator* – If the required gate movement (DU) is less than the minimum gate movement (MnMG; 0.01’ is used in the simulation), then the DU is added into an accumulator. When the gate accumulator is greater than the MnMG, then the gate should move.

3. *Hunt prevention* – Keep a record of the previous gate movement (the movement performed in the last control time step), and compare it with the calculated required gate movement in the current control time step. If the gate has moved in opposite directions and both movements are larger than a certain amount, then only move the gate a certain percentage of the calculated required gate movement in the current control time step.

4. *Fast response and quick attainment of target flow* – Immediately check to see if the target flow has changed; if it has changed, then start the response immediately, rather than waiting until the entire control time step has elapsed. Also, for the first 10 control time steps, use movements that are equal to 1.5 times the required movements; in other words, use a Reduction Factor of 0.45.

5. *Prevention of too much flow being released* – Keep track of whether the flow is rising or dropping, and by how much. Do not move the gate if, when the target flow is raised, all of the following conditions occur:
   a. a certain percentage (20%) of the required increase for the flow has been achieved,
   b. the flow keeps rising by more than a certain amount (20% of max flow) per 5-minute control time step, and
   c. the logic still calculates an upward gate movement.

   *Prevention of a shortage of delivered flow* – Use a similar logic for lowering the target flow. Do not move the gate if, when the target flow is decreased,
   a. only 20% remains of the required decrease for the flow,
   b. the flow keeps dropping by more than (20% of max flow) per 5-minute control time step, and
   c. the logic still calculates a downward gate movement.

6. *Water depth dropping upstream of gate* – in some situations, the gate movement in the previous control time step may have been an upward gate movement, but the flow in this control time step is less than the flow in last control step by more than a certain amount (such as 2-5% of max).

   In this case, use a Reduction Factor of 1. Also, do not move the gate if:
   a. the flow is dropping by more than 2-5% of max,
   b. the current flow is dropping with less then 10-20% of max remaining above the target flow, and
   c. the logic still calculates a downward gate movement,
7. Water depth rising upstream of gate – in some situations, the gate movement in the previous control time step will have been a downward gate movement, but the flow in the current control time step will be higher than the flow in last control time step by more than 2-5% of the max.

In this case, use a Reduction Factor of 1. Also, do not move the gate if:
   a. the flow is rising by more than 2-5% of max,
   b. the current flow is less then 10-20% of max lower than the target flow, and
   c. the logic is still calculating an upward gate movement.

These added heuristic adjustments are necessary for the improvements that were shown in Figure 5-4, which cannot be obtained simply by adjusting the Reduction Factor.

Normally there is an optimum combination between the RF constant and the control time step; for example, the (RF = 0.4 & Control Time Step = 1 minute) may have a similar control effect of using (RF = 0.28 & Control Time Step = 2 minutes), which depends on the pool’s size and length, the flow rate measurement device, the check structure, etc. It may take several hours or a day to find this kind of optimum combination.

An important point to note is that, no matter how long the control time step, a control action must always be based on the average of the measurements taken by the sensors in the 1 or 2 minutes before the present control action.

Special Considerations for Algorithm Tuning

Limitations to the Vertical Gate Opening

Sometimes the control algorithm may compute gate movements that actually exceed the physical or control capabilities of the gate in the field. To prevent this, ITRC must add some additional checks and balances to the control equation that consider these limitations.

The first thing to consider is the individual gate’s physical capabilities. These will set limitations on the gate’s minimum and maximum vertical opening within each control time step. Obviously, an algorithm control calculation that requires the gate to move beyond, or faster than, its physical limitations should not be executed. Therefore, the control algorithm’s calculations for gate movement should be checked against some set values regarding minimum and maximum vertical opening and maximum vertical movement rate. If a calculated movement exceeds one of these set values, the action will not be performed or will only perform according to its limit.

Normally, if a gate can move in very precise, small increments, the minimum vertical gate opening can be set to a distance as small as 0.01’; otherwise the minimum vertical gate opening will need to be set as the minimum movement that the gate can physically achieve such as 0.02’–0.05’, etc. The set values for the gate’s maximum vertical opening within each control time step are more dependent upon the actual gate capabilities and must be reconsidered for each gate. For example, at the Delta-Mendota Canal (DMC) and Umatilla...
Stanfield Branch-Furnish Canal (USBFC), the physical maximum gate movement is 1.2’/min, and the algorithm set for the maximum gate movement within each control time step is 0.6’, but at other locations this value could be physically too small or too big, according to the limitations of the gates.

Though the gate can be moved at a quicker maximum movement rate such as 1.5’/min, ITRC would still prefer to set the gate movement at no more than 0.6’-0.8’ per each control action. This means that even though the control time step is 5 minutes and the gate can be moved to a maximum of 7.5’ within each control period, for safety reasons ITRC still lets the gate move no more than 0.6’-0.8’ per each 5-min control action.

Although a maximum vertical gate opening should be set for all gates, the movement of undershot gates is actually more limited by its control capabilities. To provide smooth control, the bottom of an undershot gate must remain below the upstream water surface elevation at all times so that there are no sudden shifts between submerged and unsubmerged conditions. (For a radial gate, if the downstream water level > ½ × vertical gate opening, the gate is “submerged” or “flooded” or “drowned”; if the downstream water level ≤ ½ × vertical gate opening, the gate is “unsubmerged” or “free-flow”.) If the bottom of an undershot gate rises above the upstream water surface, it will switch from orifice control to weir control and provide discontinuous results.

Since water level conditions are constantly changing, it is not enough to say that the gate may not rise above the water surface – the water level may suddenly drop, leaving the gate high and dry. Therefore, a “buffer zone” must be observed between the gate’s vertical opening and the water level. This buffer zone must be large enough to ensure that the gate will remain below the water surface, even under changing water level conditions.

When working on the control algorithm for DMC, the original conditions set by ITRC required that the gate remain only 0.1’ below the upstream water surface elevation. When the computational time step for the algorithm was changed from 10 seconds to 1 second, this condition was too low and did not prevent the gate from going above the upstream water surface elevation. With a 1-second computational time step, when the gate rose to 0.1’ below the u/s water surface elevation, in the next moment the water surface elevation would fall below the gate opening. This problem can be seen in Figures 5-6 and 5-7.
As Figure 5-6 shows, a 0.1’ buffer zone is too small. ITRC now requires that underflow gates should not be allowed to open higher than 0.4’ below the upstream water level. Figure 5-8 shows the improved control effects at DMC when a 0.4’ buffer zone was used with the 1-second computational time step.
Figure 5-8. Elimination of gate control problem for the 1 sec. computational time step simulation when the gate must stay 0.4’ below the upstream water surface elevation – the correct limitation

Limiting the vertical opening of an undershot gate to 0.4’ below the current water level surface takes precedence over the required vertical gate opening calculated by the control algorithm. For example, a radial gate used for upstream control is 11’ wide × 9’ high. Its maximum mechanical vertical gate opening is 8.5’. If the upstream target depth is 7.3’, then even if the vertical gate opening calculated according to the PIF equation is 7.8’, the gate will still only move to a 6.9’ (7.3’–0.4’) height. Limiting gate movement to the maximum and minimum vertical gate openings and the maximum vertical movement rate also takes precedence over the control calculations.

In summary, ITRC limits the vertical movement of gates according to the following guidelines:
- The maximum vertical movement of the gate within each control action should not be exceeded.
- If the computed gate movement is less than the minimum change in vertical gate opening, don’t move the gate (see next section).
- The gate should not move below the sill height.
- The movement of any gate should not exceed its physical limits (maximum vertical gate opening).
- If the computed gate position for an undershot gate is less than 0.4’ below the upstream water surface elevation, the new gate position shall nonetheless be limited to 0.4’ below the upstream water surface elevation. (This applies for both upstream and downstream flow control.)
Tuning Algorithms for Automated Canal Control


- If the computed vertical gate height for an overshot gate is higher than 0.2’ above the upstream water surface, the new gate height shall be limited to 0.2 above the upstream water surface, which means never let the gate go above 0.2 higher than the upstream water surface.

**Use the Gate Accumulator to Zero-In On the Water Level to Target**

The gate movement accumulator (GA) register constant in CanalCAD can be used to help minimize water level deviation. The gate movement accumulator works in conjunction with another CanalCAD register constant called MnMG, for “minimum movement of gate”, to help eliminate excessive, small gate movements. MnMG represents the smallest gate movement that will be allowed.

When the absolute value of a required gate movement calculated by the control algorithm is smaller than MnMG, it is accumulatively added in the GA register and the GA absolute value is compared to MnMG. Also, if the absolute value of the calculated movement is smaller than MnMG, the movement is not executed. Instead, its value remains in the GA register. If the next time the absolute value of a gate movement calculated by the control algorithm is still less than MnMG, it is added to the value already in the GA and the absolute value of this total is compared to MnMG again. On the other hand, if the next time the absolute value of a gate movement calculated by the control algorithm is greater than or equal to MnMG, it will be executed and GA will be reset to zero. In other words, the GA is a kind of combined sum of continuously calculated gate movements whose absolute value is less than MnMG. As before, if the combined movement value is smaller than MnMG, the movement is not executed and the value waits in the GA for the addition of the next calculated movement.

If the absolute value of the GA (which can be either positive or negative) is larger than MnMG, the gate will be moved by the accumulated amount in the GA (not the amount of the last calculated gate movement). At this point, the GA register will be re-set to zero and the cycle will start over.

By adding small gate movement calculations together, the gate is moved less frequently. Since the values entered into the GA are additive, a small positive gate movement and small negative gate movement can cancel out each other, resulting in smoother overall gate control. This results in better water level control.

As can be seen in Figure 5-9, using the gate movement accumulator can actually help “zero-in” on the target water level.
In Figure 5-9, the improved water level control was achieved by moving the gate when $\text{GA} \geq \text{MnMG}$. However, Figure 5-10 shows that this gate control was still unsatisfactory, with frequent small zigzag gate movements of 0.01’.

The small zigzag gate movements shown in Figure 5-10 can be greatly decreased by accumulating the GA to $2\times\text{MnMG}$, rather than $1\times\text{MnMG}$, and then moving the gate by $\frac{1}{2}$ of the GA amount. The improved effects can be in Figure 5-11.
Tuning Algorithms for Automated Canal Control

**Figure 5-11.** When $|GA| \geq 2 \times MnMG$, gate is moved by $\frac{1}{2} \times GA$ – results in improved gate control

Notes:  
- GA is not defined in the .ccd file “regcon”, register constant, list, but is a regcon within the algorithm so that its value is retained between possible control actions for each gate.

- The condition of “when $|GA| \geq 2 \times MnMG$, then move the gate by $\frac{1}{2} \times GA$” is only suitable for the $MnMG = 0.01'$. If the physical $MnMG$ can only be as small as 0.05’ like the USBFC overshot gates, ITRC still recommends using the condition of “when $|GA| \geq 1 \times MnMG$, then move the gate by $GA$”. It can make a huge difference to wait until the GA gets above 0.1’ and then move by $\frac{1}{2} \times GA$, or to wait until the GA gets above 0.05’ and then move by $GA$.

**Using a Bigger Ramp Time for Smooth Response to Big Turnout Flow Changes**

Sometimes changing the ramp time can smoothen out drastic water level responses to big turnout flow rate changes, though not completely. “Ramp time” is the time that is allowed to complete a flow rate change to the turnout or head flow schedule of a canal.

The Sutter Mutual Portuguese Bend Canal uses PIF control logic based on water depth about $\frac{3}{4}$ of the way (1800’) downstream of the pumping plant. Control actions are implemented once a minute to maintain the 3.4’ target depth, using the Variable Frequency Drive (VFD) pumps in the pump house. Although the water depth at the target point was controlled well, at times the water depth at this location would reach higher than the canal banking lining. This problem can be solved by changing the ramp time to accommodate big flow changes. If a large turnout flow rate change is spread out over a longer period, the flow won’t build up quickly enough to cause the drastic changes in water level.

The initial ramp time at Portuguese Bend was 2 minutes. To improve the water level response, the ramp time was changed into a function of the desired flow rate change:

$$\text{Ramp time} = \text{the absolute flow change} \times (5 \text{ min/10 CFS flow change}).$$
In other words, the ramp time is increased by 5 minutes for every 10 CFS of desired flow rate change. This means that a desired flow rate change of 35 CFS would have a ramp time of 17.5 minutes (35 CFS \times (5 \text{ minute}/10 \text{ CFS})). A flow rate change of 5 CFS would have a ramp time of 2.5 minutes.

Figure 5-12 shows the control effects of using the original ramp time of 2 minutes and a ramp time that is a function of the flow rate change.

![Figure 5-12](image)

**Figure 5-12.** Comparison of water level response at Sutter Mutual’s Portuguese Bend using ramp time = 2 minutes and ramp time = \(|\text{flow change}| \times (5 \text{ minutes} / 10 \text{ CFS flow change})

However, sometimes increasing the ramp time may not be realistic if the farmer does not like the restrictions on how the pump should be turned on.

**Using Control Based on the Water Surface Elevation for Pools with a Variable Invert Bottom Slope**

For a canal in which the pool’s invert bottom slope changes over its length, the control effects should be based on water surface elevation, rather than water depth. This is especially true if a weighted average of the water depths at the head and end of the pool is used to maintain a target depth at an intermediate target point. When the invert slope varies over the length of the pool, and/or there is a long siphon such as 2000’-3000’ ft along the pool, the weighted estimate and actual depth at the target point can be quite different.

An example of a pool with a variable invert slope is Pool #1 at Patterson Canal. Its canal invert profile is shown in Figure 5-13. The invert profile for Patterson Pool #1 includes
Stations 1+00 to 74+50. Note the difference in the invert slopes between stations 1+00 to 26+00 and 27+00 to 74+50. The result is an inconsistent water depth within this pool.

![Canal Invert Profile](image)

**Figure 5-13. The Patterson Canal invert profile for Pool#1, where there is a slope variation**

For this pool, the control was trying to maintain a water depth of 4.65’ at a target point 75% downstream of Pumping Plant #1. Below is a description of the differences in control effects based on the water depth and the water surface elevation, also illustrated in Figure 5-14.

**Control Based on Water Depth**

The pink and purple lines in the lower part of Figure 5-14 show the intermediate control based on water depth. Note that the estimated water depth (purple), based on the bival method of using the weighted average of the water depths at the beginning and end of the pool matches the 4.65’ target depth. However, the actual resultant water depth (pink) is less than the target – a result of the slope variation within the pool. The difference between these two lines (the estimated and the actual water depths) indicates that had the water depth been used for control, an algorithm “target” depth of about 5’ would have been required for the actual depth to match the true desired depth of 4.65’.

**Control Based on Water Surface Elevation**

The green and blue lines in the upper part of Figure 5-14 show the intermediate control based on water surface elevation. Note that the actual water depth (green line) fluctuates about the target water surface elevation.
Calculated water elevation based on WSE – appears correct

Actual water level based on WSE – confirms the calculated value

Calculated water level based on WD – appears correct

Actual water level based on WD – disproves the calculated value

Figure 5-14. Patterson pump #1 control effect comparison between those based on water depth and on water surface elevation

From these results it is clear that using the weighted bival of the head and end pool water depths for control purposes can result in a discrepancy between the calculated and the actual water depth, particularly when the pool has a variable invert slope, though the calculated can well represent the trend of actual level.

In cases of variable slope, control of pumping plant flow or gate movement based on water surface elevation can produce superior matches between calculated and actual values – if the elevations have been determined according to a good, comprehensive survey.

How to Move Multiple Parallel On-Site Gates

The number of parallel on-site gates has ranged from 1 to 4 for the projects that ITRC has worked on. The rules of how to move the gate(s) are generally seen as either “in series” or “parallel,” which can be described as follows:

If, for example, there are three gates on-site and the required upward gate movement is 0.5’,

1. In Series: first Gate 1 will be moved up by 0.5’/3 and stopped, then Gate 2 will be moved up by 0.5’/3 and stopped, and next Gate 3 will be moved by 0.5’/3 as well.
2. In Parallel: the three on site gates will be moved simultaneously by 0.5'/3 each. The power would not be exhausted as much when moving in series as in parallel. Some logics even give the option for the operator to select moving gates in parallel or in series, or even allow the operator to choose a few gates rather than moving all in series or parallel.

For achieving a target position, some integrators apply a “minor adjustment” logic, which means that they use a logic to move a gate to a position that is close to the target, and then adjust the gate two or three times up and/or down to achieve the exact target position. The interval between several adjustments is about 10 seconds, and they often set the control time step at 2 minutes. This adjustment logic is applied to all the on-site gates, and does not help to achieve a good and tight control but rather creates a wave along the whole canal.

ITRC uses a rule for gate movement that is different from those listed above, which is to always raise the lowest gate or lower the highest gate out of those available on-site. For example, if there are four on-site parallel gates with respective openings of 0.1’, 0.12’, 0.08’, and 0.2’, with the last three gates available and the calculated required upward movement is 0.5’, then the logic can always pick the third gate (at 0.08’) and drive it up by 0.5’. In the next time step, if the calculated upward movement is 0.35’, then the logic can always avoid the unavailable gate to pick the second gate (at 0.12’) and drive it by 0.35’. A corresponding operation can also be used for lowering the gates.

ITRC’s rule for gate movement can make the on-site gates catch up with each other so that the opening difference between any two of the on-site gates is bigger than the maximum gate movement within each control time step. This rule has been tested on two on-site parallel gates at Highline Grand Junction, and several areas in Tulare ID.

**Considerations for Working with Equipment and Integrators**

**PLC and Sensor Constraints**

ITRC has encountered several PLC and sensor constraints. On the PLC side, we have learned that many PLCs require a significant part of a minute just to run through various checks of equipment, to read values from sensors, and to communicate with SCADA systems. Furthermore, the manufacturers are often unable to tell us how much time is required, and it is common that most of the PLCs use their own proprietary software, which is un-transferable to other PLCs. Because our simulations should duplicate the actual computations, this is problematic. We have also found that some brands will take 60 sensor values per minute and provide an average, but in Grand Junction for example, those 60 sensor values of upstream and downstream water level measurements are all read at a rate of 0.3 seconds within the last 18 seconds of a 90-second control time step, and these work fine.

For applications, we now insist on redundancy of key items. Specifically, we state that all of the key sensors be duplicated, plus the sensors must be wired into different power supplies and A/D converter modules in the PLCs. Moreover, ITRC has created a series of specifications for SCADA systems, PLCs, sensor leads, radio & antenna, HMI, etc. We
assume that it is not a question of "if" a sensor will fail, but a question of "when". Although there are numerous techniques for using software to check for problems with single sensors, we have found that this adds a tremendous complexity to the programming of the PLC that is unnecessary if redundancy exists.

On the sensor side, there are the classic problems with accuracy, calibration, and resolution. But a new challenge is presented when one uses an ADFM or similar device to measure flow rates at the head of a canal (for control purposes). Figure 5-15 shows that there is tremendous noise in their signals even though the meter itself uses some unknown averaging techniques. We have examined numerous filtering techniques, but in the end we have concluded that we need at least 10 minutes of continuous readings after the meter has been deployed and is giving an averaged reading once per minute before we can use an average value for a control decision. This complicates the control of headworks on some canals. In contrast, using a Replogle flume to measure flows at the head of a canal has advantages:

(i) little or no random noise in the signal,
(ii) inexpensive redundancy of the water level sensor, and
(iii) because the Replogle flume is a critical flow device, the new flow rate stabilizes very rapidly, so it is easy to determine how much to change the flow control device (this is very important for control).

![Figure 5-15. ADFM and filtered signals at Headgate Rock Dam – CRIT irrigation project, Arizona](image-url)
PLC Programming by Integrators

A big surprise has been the complexity of dealing with integrators. It appears that each integrator company has been quite independent. Their procedures for documentation of programming, their neatness of organizing wiring and panels, their usage of programming languages, and their exposure to PI algorithms and calibration and even alarms for canal automation are very different. This means that nothing can be taken for granted – even if an integrator can list numerous completed projects.

Three items are of particular concern to us:

1. A good integrator will always understand hardware, installation, communications, and programming quite well. But it is rare that an integrator is familiar with modern canal control algorithms, and how they are tuned within a simulation model. It was found that the integrators typically used the Proportional logic with empirically chosen constants, or may even have programmed the logic in the Littleman controller method, which means they would move the gate by a certain amount of time proportional to the water level error. This can be a problem if the integrators take unwarranted liberties in the programming of the control algorithms that we supply, as well as with the tuning constants that we provide.

2. Integrators sometimes embed numerous checks into their code with various hidden constants (of their own selection) that can shut down a gate or pump operation. The irrigation district operators (i) do not know these constants exist, (ii) do not know how to access them, (iii) must generally personally visit the PLC to change the constants, and (iv) do not really understand how the constants should be changed. We believe all constants and alarms should be transparent and changeable from within the office via the SCADA system. A portable PC with a copy of the office HMI (Human-Machine-Interface) software can be used in the field to change constants if it is desired to make in-field adjustments.

3. The "control algorithm" for a gate (the algorithms that are published in irrigation literature) may only occupy 10% of the total programming that most integrators put into the PLC. The remainder of the programming handles numerous checks of equipment and sensors, consideration of gate inertia, and other factors including unnecessary alarms such as “hi”, “hi-hi”, and “hi-hi-hi” alarms, where we normally say that only one high alarm is enough.

In the past, ITRC provided the control logic in a flow chart format created inside SmartDraw, which is a non-executable Word-similar drawing program. We insisted that the integrator should program the algorithm exactly according to the ITRC-provided flow chart. However, numerous extra minor points were still being programmed in by integrators. These minor points sometime play a big role in the control effects. Some examples of this include:

a. Wrong units of measurement. ITRC emphasizes that all the sensors should be calibrated and displayed in floating point values in units of feet; however, in some cases all the sensors were calibrated and displaying in integer values of 1/100th inch. Inside the algorithm, only when these values were changed to floating point ft. values
were they used in the PIF equation. Correspondingly, all the settings for the level and gate position calibrations and alarms are then required to be input in $1/100^{th}$ inch rather than in the units of ft.

b. **Redundant alarms.** The alarms that were found in the programming are combinations of ITRC-designed alarms (which are only for the selected primary sensor with high & low alarms) and the integrator’s own designed alarms (which are for all the sensors with a hi-hi & hi and low-low & low for each). These inevitably add some redundant alarms and would make the HMI have too many alarm options. Sometimes having too many alarms can bother the operator, who in turn may input a very high and very low value so that the alarm will never click on or they may disable the alarm so that even though the alarm is kicked on, it would not dial out.

c. **Incorrect speed limitations.** The internal VFD pump speed was clamped to a max. speed of 880 RPM, which was referenced by the max. speed of 890 RPM that is shown on the VFD pump nameplate. However, that VFD pump was actually designed to operate at a maximum of 945 RPM, and the ITRC flow chart used 945 RPM as the maximum and did not provide for the internal clamps and other restraints on it. It was understandable that the clamp was designed to protect the pump from running at its max limit, but it hampers the internal switch logic used to turn on another pump and ramp down the VFD’s speed.

d. **Adjustments to gate movement.** ITRC logic moves the gate to the desired location with the understanding that the gate will overshoot a bit when it moves down because of inertia and/or brakes, and the gate may stop exactly right at or a little bit lower than the right position when it goes up because of the resistance of the gate and the water. Despite this, ITRC did not suggest any “soft” corrections such as driving the gate down a little bit overshot and bringing it up, then letting the gate stop within a dead band of the target position when it moves up. Unfortunately, these kinds of soft adjustments were designed by the integrator and programmed into the logic.

These kinds of minor discrepancies between the integrator programming and the ITRC-provided logic make the ITRC review of the integrator programming important to ensure that the final control logic is as the ITRC CanalCAD model anticipated. There is no doubt that the integrator’s review of ITRC’s logic and flow chart is important to achieve a successful canal control project. However, this work is time-consuming and must be done at every single project, even though the ITRC control logic is the same as that used on other projects with the same type of control.

To solve a few of these problems, ITRC has recently begun providing the control logic code drawn in a flow chart with formatted structured text using ISaGRAF (explained in the next section). The ISaGRAF control code is then provided to the integrator, and the integrator only adds a few separate modules for their own communications or other functions that are insignificant to the control logic. In this way, the implementation work by different parties is separated very distinctly.
Control Code Programming in ISaGRAF

ISaGRAF is a product of ICS Triplex. The ISaGRAF program is consistent with the standards of IEC 61131-3 industrial control languages, and is sold to PLC manufacturers, as is Hibeam, a kind of HMI. ISaGRAF was originally introduced in 1990 for bridging the gap between microcomputer systems and PLCs; currently, it is suitable for both centralized and distributed control systems that support 32, 64, 128, 256, or unlimited input/output points.

The relationship between ISaGRAF and the PLC is shown in Figure 5-16.

![Figure 5-16. Relationship between ISaGRAF and the PLC](image)

The driver and runtime module are normally obtained from the PLC manufacturer at an insignificant or no cost. The ISaGRAF license fee may also vary depending on how many PLCs need to be run.

ISaGRAF is independent of hardware and software during the control code development. The great benefits of it are:

a. It is easily transferable between different brands of PLCs that accept the ISaGRAF programming.

b. The simulation tool inside ISaGRAF enables the programmer to examine how the code is running and what the value/state for each variable is when the actual PLC is not connected. This is a powerful tool since it provides the capability to run and debug the control code before the PLC controller is chosen.

Most of proprietary PLC software programming tools are incapable of doing “a” and “b” above. For this reason, ITRC has chosen ISaGRAF as our programming tool for all canal control-related purposes using PLCs.
Overview of ITRC Control Algorithm and ISaGRAF Modules of Flow Chart in Structured Text Language

The modular programming inside ISaGRAF for a typical canal control with different control schemes such as upstream water level control, downstream water level control, flow control, etc. is shown in Figure 5-17.

Taking one general module as an example, the ISaGRAF programming style of flow chart in structured text is shown in Figure 5-18.
In general, ITRC is responsible for providing:

- ISaGRAF control code for the proposed control logic, and
- Correct control actions in both manual and auto movement mode and the correct alarm generation when needed.

The integrator is responsible for:

- sensor selection and installation;
- PLC wiring and labeling;
- radio and repeater;
- alarm auto-dialing;
- HMI design.

The line that distinguishes the ITRC-provided ISaGRAF control code and the integrator’s work involves the assigned registers for each site. ITRC provides to the integrator a full explanation of the tags and the assigned registers and their associated recommended values that need to be designed and displayed in HMI. The integrator ensures that the ITRC-listed HMI variables that are assigned with the associated registers can be either input through or displayed in HMI either through radio and/or a repeater.

The key in such a co-operation is a clear line between tasks and responsibilities. Using this approach, ITRC has completed control logic programming in ISaGRAF and it has passed the bench-testing for upstream water level, downstream water level, flow control and some special spill control situations for up to three on-site parallel check structures. One customer has already finished transferring the previously programmed Ladder logic to ISaGRAF for the upper part of the canal’s upstream level control, and finished the programming in ISaGRAF for the lower part of the canal’s automation with some additions of customized changes such as downstream level control and flow control for gates and pumps, both with and without Variable Frequency Drives (VFDs). ISaGRAF has facilitated the transfer of the control code between controllers, and the development of the control code for a new system takes much less time. This, combined with the revised roles of ITRC and the integrator, makes the field implementation of the control code much quicker and more efficient.

ITRC anticipates that the actual implementation of the control logic, which covers the sensor calibration and field debugging/testing of the control logic, should only take both ITRC and the integrator less than a week. In the spirit of teamwork, ITRC makes every effort to cooperate with integrators to achieve these goals.
REFERENCES


