

# Automatic Tuning of PI Controllers for an Irrigation Canal Pool

X. Litrico\*, P.-O. Malaterre†, J.-P. Baume‡, P.-Y. Vion§, J. Ribot-Bruno¶

Paper published in the *Journal of Irrigation and Drainage Engineering*, Vol. 133, No. 1, January/February 2007

## Abstract

The paper presents a method to automatically tune decentralized Proportional Integral (PI) controllers for an irrigation canal pool. The Auto Tune Variation (ATV) method is based on a relay experiment, which generates small amplitude oscillations of the canal pool. The ATV procedure can be used to get the Integrator Delay model parameters of a canal pool, which in turn can be used to tune a Proportional Integral controller using classic rules, or other rules such as the ones proposed by Litrico and Fromion (2006). This method does not require advanced automatic control knowledge and is implemented in SIC software, developed by Cemagref, which also incorporates a Supervisory Control And Data Acquisition (SCADA) module for real-time control. The ATV method is evaluated by simulations and experiments on a real irrigation canal located in the South of France, for local upstream, local downstream and distant downstream controller tuning.

## Introduction

Most regulated irrigation canals are controlled by Proportional (P), Proportional Integral (PI) or Proportional Integral Derivative (PID) controllers. This is not surprising, since as stated by Aström and Hägglung (1984), “the vast majority of the regulators used in industry are of the PID type”. However, these regulators with only one (P), two (PI) or three parameters (PID) are not so easy to tune. Several different methods have been proposed for tuning PI controllers for irrigation canals, see e.g. Schuurmans et al. (1999b); Baume et al. (1999); Weyer (2002); Clemmens and Schuurmans (2004); Litrico and Fromion (2006); Litrico et al. (2006). All these methods are based on a model of the canal, which is not always easy to obtain. Indeed, one needs an extensive measurement campaign to calibrate a full Saint-Venant model, or a large number of experiments to identify linear models for control (Weyer, 2001). The obtained model can be linear, such as the Integrator Delay (ID) model (Schuurmans et al., 1999a; Litrico and

---

\*Researcher, UMR G-EAU, Cemagref, B.P. 5095, 34196 Montpellier Cedex 5, France, e-mail: [xavier.litrico@cemagref.fr](mailto:xavier.litrico@cemagref.fr)

†Researcher, UMR G-EAU, Cemagref, B.P. 5095, 34196 Montpellier Cedex 5, France. e-mail: [pierre-olivier.malaterre@cemagref.fr](mailto:pierre-olivier.malaterre@cemagref.fr)

‡Hydraulic Engineer, UMR G-EAU, Cemagref, B.P. 5095, 34196 Montpellier Cedex 5, France. e-mail: [jean-pierre.baume@cemagref.fr](mailto:jean-pierre.baume@cemagref.fr)

§Hydraulic Engineer, UMR G-EAU, Cemagref, B.P. 5095, 34196 Montpellier Cedex 5, France. e-mail: [pierre-yves.vion@cemagref.fr](mailto:pierre-yves.vion@cemagref.fr)

¶Assistant Engineer, UMR G-EAU, Cemagref, B.P. 5095, 34196 Montpellier Cedex 5, France. e-mail: [jose.ribot-bruno@cemagref.fr](mailto:jose.ribot-bruno@cemagref.fr)

Fromion, 2004) or nonlinear, such as the full Saint-Venant model. However, none of these models include real-life features such as communication delays or actuators/sensors limitations. These limitations do apply when the controller is implemented on the real canal, and they may hinder the functioning of the controlled system, and even destabilize it. It would therefore be interesting to have a simple method to tune PID controllers based on the real system, including its limitations, communication delays, and nonlinearities.

Another objective is to have a 'plug-and-play' control method, that would enable the user to use a controller that automatically computes its parameters in a single experiment, and starts to control the canal directly after this experiment, without needing advanced automatic control knowledge.

The Auto Tune Variation (ATV) method is a classical automatic tuning method first proposed by Aström and Hägglung (1984). This method identifies important characteristics of the response of a monovariate process using a simple relay experiment. It has been applied in many industrial processes (Hang et al., 2002), in conjunction with tuning rules based on the ultimate cycle analysis. The relay experiment identifies with a single straightforward experiment the ultimate cycle parameters (ultimate gain  $k_u$  and ultimate period  $T_u$ ) of a monovariate process represented by a linear transfer function. Once these parameters are obtained, any classical tuning rule based on the ultimate cycle can be used to obtain a Proportional (P), PI or PID controller.

We show in this paper that a relay experiment also directly identifies the parameters of an ID model for a canal pool, and we derive the relation between the ID model parameters and the ultimate cycle parameters. We analyze classical tuning rules proposed in automatic control literature, such as the well-known Ziegler-Nichols rule (Ziegler and Nichols, 1942), by evaluating their gain and phase margins in the case of a dimensionless ID model, following the approach developed by Litrico and Fromion (2006). We also propose a new tuning rule for P, PI or PID controllers of an irrigation canal pool based on the ultimate cycle parameters, to ensure desired gain and phase margins.

The method is first implemented and simulated with SIC (Simulation of Irrigation Canals), a computer program developed by Cemagref solving Saint-Venant equations for irrigation canals which incorporates an ATV control module (Malaterre and Baume, 1997; Malaterre, 2006). It is also validated on a real irrigation canal, the Gignac Canal located in the South of France 40 km north-west of Montpellier, for local upstream control, local downstream control and distant downstream control of different canal pools. These experiments are done using the real-time Supervisory Control And Data Acquisition (SCADA) module of SIC, enabling the real-time communication between the hydraulic simulation software SIC and real-time measurements on the canal.

## Automatic Tuning for a Canal Pool

### Description of the Ultimate Cycle Method

The objective of the ultimate cycle method is to determine the critical point, i.e. the process gain and frequency leading to a phase lag of  $-180^\circ$ . This point has been employed to set the parameters of PID controllers since the advent of the Ziegler-Nichols (ZN) rule (Ziegler and Nichols, 1942). The critical point is traditionally described in terms of the ultimate gain  $k_u$  and the ultimate period  $T_u$ , obtained from ultimate cycle analysis. In this case, the system is connected in a feedback loop with a proportional controller (see Fig. 1, where  $u$  is the control action variable,  $y$  the controlled variable and  $r$  the set-point variable). The variable gain

of the controller is then increased or decreased until the process output presents undamped oscillations, which correspond to the ultimate cycle. The ultimate cycle parameters are the period of oscillations, denoted by  $T_u$ , and the proportional gain, denoted by  $k_u$ . This method is time consuming, since the ultimate gain is obtained via a trial-and-error procedure. The gain  $k$  has to be decreased if the amplitude of oscillations is increasing, and it has to be increased if the amplitude of oscillations is decreasing.

Moreover, the ultimate cycle method has the disadvantage to put the process at its limits of stability, which is generally not advisable, especially in the case of irrigation canals. In many cases, management constraints forbid the implementation of the ultimate cycle analysis, in order to prevent large oscillations of the water level in the canal, which may lead to overtopping.

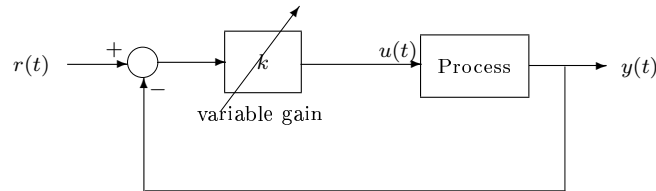


Figure 1: Ultimate cycle feedback system

Typical responses to the ultimate cycle analysis method are shown in Fig.2, where the gain  $k_u$  corresponds to the ultimate gain, leading to constant amplitude oscillations. We observe in this figure that the output error and the input variable oscillate with a constant amplitude. If the amplitude of the output error is equal to  $2a$ , then the amplitude of the control variable is equal to  $2k_u a$ . The period of oscillation is  $T_u$ . In the case of a dimensionless Integrator Delay system of transfer function  $e^{-s}/s$ , it is a classical derivation to compute the ultimate gain  $k_u = 1.57$  and the ultimate period  $T_u = 4$ .

### Description of ATV Method

The relay feedback auto-tuning proposed by Aström and Hägglung (1984) was one of the first to be commercialized for tuning of PID controllers in industry. It has since remained attractive owing to its simplicity and robustness. In this method, the process to control is connected in a feedback loop with a relay, as shown in Fig. 3. Contrarily to the ultimate cycle method, the ATV method is a straightforward procedure where the manager can choose the maximum value  $d$  of gate opening variation for the relay experiment. The relay is a simple nonlinear element that changes the input  $u$  to  $+d$  when the output error  $e = y - r$  becomes negative, and to  $-d$  when the output error becomes positive. This leads to an amplitude of  $2d$  for the variation of input variable.

It is therefore very easy to implement on a real canal equipped with SCADA, since the gate has to be opened or closed according to a measured water level. With the usual assumption of negative feedback, the relay amplitude  $d$  is positive when dealing with downstream control and it is negative when dealing with upstream control.

The ATV method therefore appears to be an attractive method to directly obtain the ultimate cycle parameters without driving the canal too far away from its equilibrium point. It can be shown that under relay control, the process will oscillate with the period  $T_u$  and that the critical

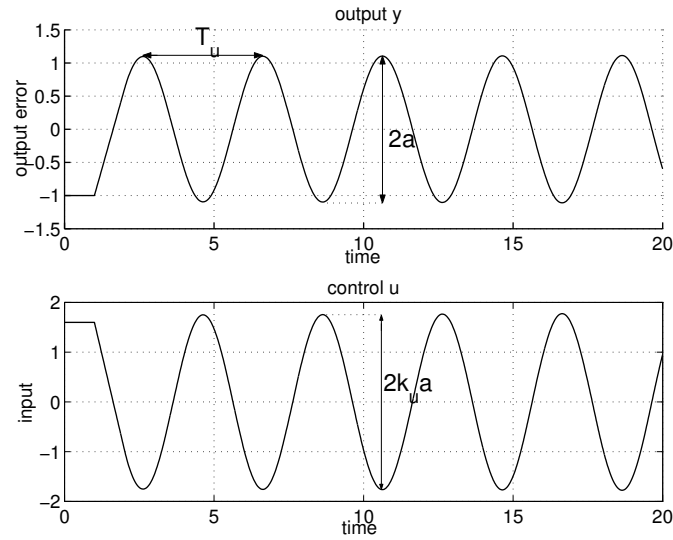


Figure 2: Dimensionless linear responses for ultimate cycle experiment on a dimensionless Integrator Delay system of transfer function  $e^{-s}/s$ .

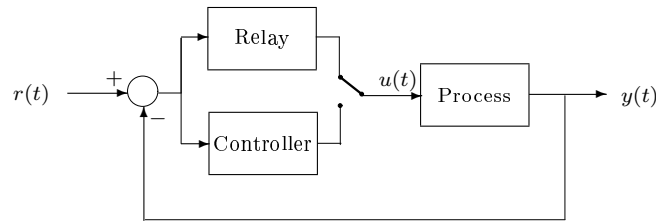


Figure 3: Relay feedback system

gain  $k_u$  is approximately given by (Aström and Hägglung, 1984):

$$k_u = \frac{4d}{\pi a} \tag{1}$$

where  $d$  is the relay amplitude and  $a$  is the amplitude of the process output. The relay control is schematized in Fig 3. Typical responses to this relay test are as in Fig. 4.

### Link with ID Model

Let us first recall the definition of the ID model. The ID model is an approximate representation of canal pool dynamics for low frequencies:

$$y = \frac{1}{A_d s} (e^{-\tau_d s} u - p) \tag{2}$$

where  $y$  is the downstream water elevation,  $\tau_d$  the delay of the canal pool and  $p$  the downstream perturbation. When the control input  $u$  is a discharge,  $A_d$  represents the backwater area, and when  $u$  is the upstream gate opening,  $A_d$  is the inverse integrator gain.

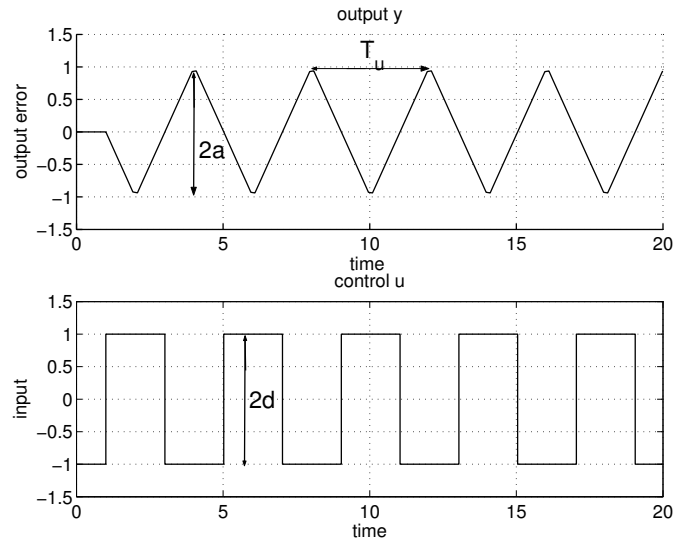


Figure 4: Dimensionless linear responses for relay experiment on a dimensionless Integrator Delay system of transfer function  $e^{-s}/s$ .

In the time domain, it corresponds to the following differential equation for the downstream water level:

$$A_d \frac{dy(t)}{dt} = u(t - \tau_d) - p(t) \quad (3)$$

This simple model has been shown to correctly reproduce the canal pool dynamics for control purposes (Schuurmans et al., 1999a; Litrico and Fromion, 2004).

An interesting feature of this model is that the relay experiment enables to directly get the ID model parameters. Indeed, in the case of a canal pool represented by an ID model, the ATV method provides the following relations between ultimate cycle parameters and ID model parameters (see Appendix I for details):

$$k_u = \frac{4A_d}{\pi\tau_d} \quad (4)$$

$$T_u = 4\tau_d \quad (5)$$

Therefore the relay experiment enables us to identify the ID model parameters  $\tau_d$  and  $A_d$ , since once  $a$  and  $T_u$  are measured,  $k_u$  is deduced using Eq. (1), and the ID model parameters are given by:

$$\tau_d = \frac{T_u}{4} \quad (6)$$

$$A_d = \frac{\pi k_u T_u}{16} \quad (7)$$

### Definition of P, PI and PID Controllers

For a proportional (P) controller, the control input varies proportionally to the output error  $e = r - y$ , where  $r$  denotes the reference output. Then, we get:

$$u(t) = k_p e(t) \quad (8)$$

A PI controller is classically defined by two parameters, the proportional gain, denoted  $k_p$ , and the integral time, denoted  $T_i$ . The control input is then computed as a function of the output error  $e$ . The continuous time domain equation of a PI controller is given by:

$$u(t) = k_p e(t) + \frac{k_p}{T_i} \int_0^t e(v) dv \quad (9)$$

In the Laplace domain, this controller has the following transfer function:

$$u(s) = k_p \left( 1 + \frac{1}{T_i s} \right) e(s) \quad (10)$$

A PID controller has an additional term, related to the derivative of the output error:

$$u(t) = k_p e(t) + \frac{k_p}{T_i} \int_0^t e(v) dv + k_p T_d \frac{de(t)}{dt} \quad (11)$$

In the Laplace domain, this controller has the following transfer function:

$$u(s) = k_p \left( 1 + \frac{1}{T_i s} + T_d s \right) e(s) \quad (12)$$

In practice, the derivative term is usually filtered in order to become less sensitive to sensor noise.

### Analysis of Classical Tuning Rules

Many different rules have been proposed to tune PI or PID controllers. We focus in this paper on the classical Ziegler-Nichols rule and other specific rules proposed for Integrator Delay systems. The classical Ziegler-Nichols rule gives a PI controller with the following parameters (Ziegler and Nichols, 1942):

$$k_p = 0.45k_u \text{ and } T_i = 0.83T_u \quad (13)$$

Expressing the ultimate cycle parameters as functions of the ID model parameters with equation (4) and (5), this tuning rule gives

$$k_p = 0.57 \frac{A_d}{\tau_d} \text{ and } T_i = 3.33\tau_d \quad (14)$$

With these PI coefficients, one may compute the corresponding gain and phase margins of the controlled system using Eqs. (43) and (44) in Appendix II, leading to  $\Delta G = 7.2$  dB and  $\Delta\Phi = 28$  degrees. The phase margin is rather low, and leads to a high overshoot.

The same analysis is performed for other classical tuning rules based on ultimate cycle analysis. Chien et al. (1952) have proposed refined rules with no overshoot for an ID model:

$$k_p = 0.6 \frac{A_d}{\tau_d} \text{ and } T_i = 4\tau_d \quad (15)$$

Using Eqs. (6–7) these parameters can be expressed in terms of the ultimate cycle parameters:

$$k_p = 0.47k_u \text{ and } T_i = T_u \quad (16)$$

Aström and Hägglund (1995) have proposed a general tuning rule for PI controllers:

$$k_p = 0.32k_u \text{ and } T_i = 0.94T_u \quad (17)$$

Table 1: PI controller tuning rules based on ultimate cycle analysis

	$k_p/k_u$	$T_i/T_u$
Ziegler-Nichols	0.45	0.83
Chien-Hrones-Reswick	0.47	1
Aström-Hägglung	0.32	0.94
Tyreus-Luyben	0.31	2.2

Table 2: PI controller tuning rules and robustness margins for a ID model

	$k_p\tau_d/A_d$	$T_i/\tau_d$	$\Delta G$	$\Delta\Phi$
Ziegler-Nichols	0.57	3.33	7.2 dB	28°
Chien-Hrones-Reswick	0.6	4	7.2 dB	32°
Aström-Hägglung	0.41	3.76	10.4 dB	33°
Tyreus-Luyben	0.395	8.8	11.5 dB	51°

Tyreus and Luyben (1992) have proposed the following tuning rule for an Integrator Delay system:

$$k_p = 0.31k_u \text{ and } T_i = 2.2T_u \quad (18)$$

Table 1 summarizes the rules based on ultimate cycle analysis for PI control of an ID model. Since we consider a canal pool represented by an ID model, one may use equations (4–5) and Table 1 to express the PI controller parameters as functions of the ID model parameters. This is given in table 2, together with the gain and phase margins of the obtained controllers.

These tuning rules lead to the time responses depicted in Fig. 5 for a dimensionless ID model, represented by a unit delay and a unit integrator gain. The dimensionless simulation results are based on a scenario including a unit step change of the downstream level set-point ( $r$  changes from 0 to 1 at time 0), followed after 30 time units by a unit step change of the downstream perturbation ( $p$  changes from 0 to 1 at time 30). The simulation is stopped after 60 time units. From this figure, one may choose a desired output response, and choose accordingly the tuning rule. However, it is not clear which is the “best” rule. Indeed, the Ziegler-Nichols rule appears to lead to oscillating response, due to the small phase margin. With a similar gain margin, the Chien-Hrones-Reswick method has a larger phase margin, and a better time response. The Aström-Hägglung method has a good gain margin, but its small phase margin leads to an oscillating response. The Tyreus-Luyben method may appear as too slow, but it ensures high gain and phase margins.

Since it is not easy to choose between these methods, we propose to use a previous work on the tuning of PI controllers for Integrator Delay models by Litrico and Fromion (2006) in order to provide a flexible tuning method for P, PI or PID controllers based on ATV method.

## New Robust Tuning Rules

### PI controller

Once the ID model parameters are known from the relay experiment, it is possible to use the results already obtained by Litrico and Fromion (2006) concerning the tuning of PI controllers for an irrigation canal based on gain and phase margins specifications. Assuming an ID model for a canal pool, Litrico and Fromion (2006) showed that in order to get a gain margin of  $\Delta G$

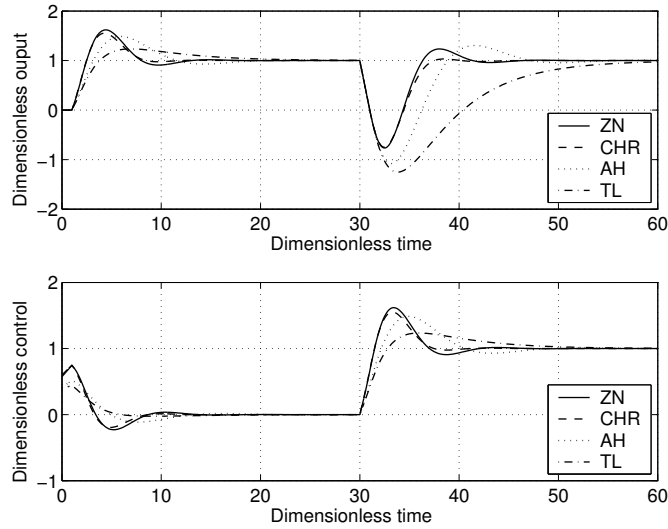


Figure 5: Dimensionless linear responses for PI control of a dimensionless Integrator Delay system with tuning rules of table 2.

dB and a phase margin of  $\Delta\Phi$  degrees, the PI controller parameters should be chosen as:

$$k_p = A_d \omega_c \sin\left(\frac{\pi}{180} \Delta\Phi + \tau_d \omega_c\right) \quad (19)$$

and

$$T_i = \frac{1}{\omega_c} \tan\left(\frac{\pi}{180} \Delta\Phi + \tau_d \omega_c\right) \quad (20)$$

with  $\omega_c$  the crossover frequency, given by:

$$\omega_c = \frac{\pi}{2\tau_d} 10^{-\frac{\Delta G}{20}} \quad (21)$$

In this way, the controllers parameters can be chosen according to time performance and robustness specifications, and not chosen from pre-specified rules (such as Ziegler-Nichols rule) that do not give to the user the ability to change the controller according to performance and robustness requirements. Using the results of Litrico and Fromion (2006) summarized by equations (19–21) and the relation between  $k_u$ ,  $T_u$ ,  $A_d$  and  $\tau_d$  given by equations (4–5), we obtain the following tuning rules for PI controllers based on a relay experiment in order to get a gain margin of  $\Delta G$  dB and a phase margin of  $\Delta\Phi$  degrees:

$$k_p = k_u \frac{\pi^2}{8} 10^{-\frac{\Delta G}{20}} \sin\left(\frac{\pi}{180} \Delta\Phi + \frac{\pi}{2} 10^{-\frac{\Delta G}{20}}\right) \quad (22)$$

$$T_i = \frac{T_u}{2\pi} 10^{\frac{\Delta G}{20}} \tan\left(\frac{\pi}{180} \Delta\Phi + \frac{\pi}{2} 10^{-\frac{\Delta G}{20}}\right) \quad (23)$$

To guarantee a well-defined proportional coefficient, the phase margin is bounded:

$$\Delta\Phi < \Delta\Phi_{\max} = 90(1 - 10^{-\frac{\Delta G}{20}}). \quad (24)$$

For classical values such as  $\Delta G = 10$  dB and  $\Delta\Phi = 43^\circ$  (Litrico and Fromion, 2006), this gives

$$k_p = 0.37k_u \quad (25)$$

and

$$T_i = 1.5T_u \quad (26)$$

## P controller

If one wishes to tune a proportional controller with a desired gain margin, following the line of Litrico and Fromion (2006) leads to:

$$k_p = k_u \frac{\pi^2}{8} 10^{-\frac{\Delta G}{20}} \quad (27)$$

In this case of a simple proportional controller, it is not possible to impose a desired phase margin. But the phase margin can be computed as a function of the gain margin, leading to:

$$\Delta\Phi = 90(1 - 10^{-\frac{\Delta G}{20}}) \quad (28)$$

Indeed, the phase margin in case of proportional control corresponds to the maximum achievable phase margin for PI control.

## PID controller

For PID controller, a similar reasoning as the one of Litrico and Fromion (2006) leads to the following equations:

$$k_p = k_u \frac{\pi^2}{8} 10^{-\frac{\Delta G}{20}} \sin\left(\frac{\pi}{180}\Delta\Phi + \frac{\pi}{2}10^{-\frac{\Delta G}{20}}\right) \quad (29)$$

$$T_i = \frac{nT_u}{2(n+1)\pi} 10^{\frac{\Delta G}{20}} \tan\left(\frac{\pi}{180}\Delta\Phi + \frac{\pi}{2}10^{-\frac{\Delta G}{20}}\right) \quad (30)$$

$$T_d = \frac{T_u}{2n\pi} \frac{10^{\frac{\Delta G}{20}}}{\tan\left(\frac{\pi}{180}\Delta\Phi + \frac{\pi}{2}10^{-\frac{\Delta G}{20}}\right)} \quad (31)$$

with  $n > 1$  and  $\Delta\Phi < \Delta\Phi_{\max} = 90(1 - 10^{-\frac{\Delta G}{20}})$ .

$n$  is a design parameter that can be tuned to specify the rapidity of the response. It is linked to the filter value of the derivative term. From our experience, average values are between  $n = 2$  and  $n = 6$ .

## Implementation Aspects

### Duration of Relay Experiment

With an estimate of the time-delay of the considered pool, it is possible to derive the time needed for the relay experiment. Let us assume that an estimate for the delay of the canal pool can be obtained, either from a real experiment, or from a simulation. Then, using Eq. (5), one gets  $T_u = 4\tau_d$ . Therefore, if one chooses a relay experiment with  $N$  oscillation cycles, the total length of the relay experiment will be  $4N\tau_d$ . From our experience,  $N = 2$  to 4 is enough to get nice regular oscillation cycles.

### Choice of Relay Amplitude $d$

The choice of the relay amplitude  $d$  is subject to opposite requirements:

- The value should not be chosen too large in order to prevent large oscillations of the water level in the canal;

- But it should also be chosen large enough to lead to visible oscillations in the canal, that are larger than the measurement noise.

The managers' experience is useful to determine the adequate value of  $d$ . In the paper, we used a value between 4 and 5 cm, but this should be adjusted to the considered gate and canal pool. As an initial value, we would suggest to choose a gate opening variation leading to 10 to 20 % variation of the discharge with respect to the initial discharge value. Larger variations may be acceptable, according to the manager's constraints.

### Beginning of Relay Experiment

If the relay experiment begins on a stabilized flow regime, the user can first change the input variable by imposing  $u = u(0) + d$  and select the stabilized output variable as a reference  $r = y(0)$ . Then the system will oscillate around the initial regime.

## Applications of ATV Method

The ATV method is applied first in simulation, then in real-life on a large scale irrigation canal located in the South of France, the Gignac Canal. In order to demonstrate the flexibility of the ATV method to tune controllers, we will test it on various pools of the canal, for different control configurations: local upstream, local downstream and distant downstream control. This paper focuses on the control of a single pool. The extension of the method to multiple pools canals is currently underway.

### Description of Gignac Canal

The experiments are performed on the Gignac Canal, located 40 km north-west of Montpellier, in the south of France (see location in Fig. 6).

The main canal is 50 km long, with a common feeder<sup>1</sup> (8 km long) and two branches on the left and right banks of the Hérault river (resp. 27 and 15 km long). The canal is concrete lined, with a rectangular cross section on the feeder and a trapezoidal one on the two branches, with average slopes of respectively, 0.00035 and 0.00050 m/m. The design flow of the canal is 3.5 m<sup>3</sup>/s. The canal has been equipped with sensors, actuators and a SCADA system, which enables the monitoring and control of four reaches in a row on the feeder above Partiteur cross-regulator and the right bank branch from Partiteur to Mas Rouvière cross-regulator. A longitudinal view of the feeder and the right bank branch is depicted in Fig. 7.

### Description of Gignac Canal SCADA System

The SCADA equipment has been designed to enable real-time monitoring and control of the canal, through a radio communication network and a real-time SCADA interface (see Fig. 8). The measured data is stored in a database, localized in the central control station.

The system has been configured to enable real-time control via simple exchange of ASCII files, in order to communicate data between the controller and the supervising system for the canal. The supervisor can generate an ASCII file (Data.txt) at each sampling time, with all the measured data of the system. This file is read by the controller, and used to compute the control variables. The control variables values are written to another ASCII file (Command.txt) which is read by

---

<sup>1</sup>A feeder is a portion of canal with no offtakes

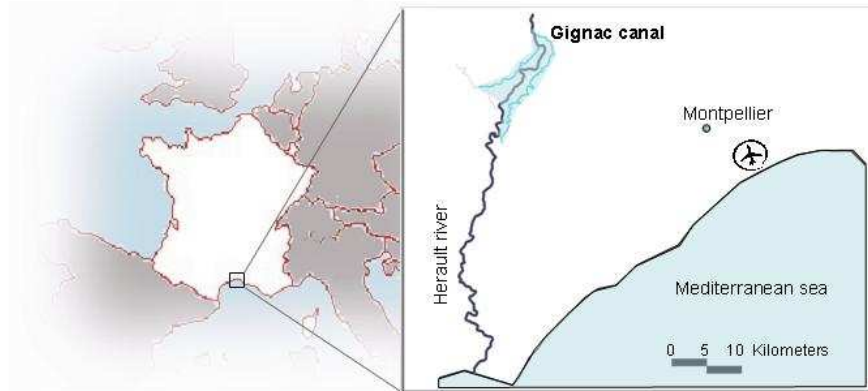


Figure 6: Location map of Gignac Canal

the supervisor and applied on the actuators. This very generic method can easily be extended to work on any supervisor system.

### Description of SIC Software

SIC software is developed by Cemagref (Malaterre and Baume, 1997; Malaterre, 2006) and implements a semi-implicit Preissmann scheme to solve Saint-Venant equations for open-channel one dimensional flow. It includes a wide variety of automatic controllers (also called control modules) that can be tested on a modelled canal. SIC offers the possibility to choose from a set of pre-programmed control modules, or to use a user-defined program, which can be written in FORTRAN or in Matlab programming language. The ATV automatic tuning method presented in the paper has been available in SIC for several years. It has been tested in simulation on a large variety of canals, with different geometry and control configurations.

### ATV control module of SIC

The SIC interface for ATV method is depicted in Fig. 9. Various tuning methods can be selected to design the controller based on the ultimate parameters obtained from the relay experiment. The user can also specify the value of the relay amplitude and the number of oscillation cycles before determination of the ultimate cycle parameters and controller implementation.

### SCADA interface of SIC

The SIC interface for SCADA real-time control is depicted in Fig 10. The values of the measured variables, controlled variables and control action variables according to the definition of Malaterre et al. (1998) are displayed in real-time. The left-hand screen displays the values measured in the field (absolute values expressed in some convenient units), the right-hand screen

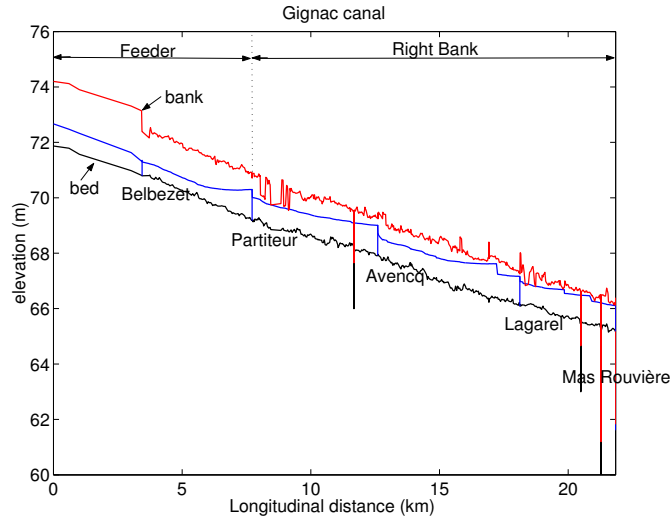


Figure 7: Longitudinal view of Feeder and Right Bank of Gignac Canal

displays the values used in the SIC software (which can either be deviations from initial values or absolute values, and are expressed in S.I. units). This screen is updated at every sampling instant, with the new values of measured variables, controlled variables and control action variables, which are computed with any controller chosen from the list of control modules available in SIC.

Therefore, it is possible to use the SIC software first as a simulator, to test and tune regulators, and second to use the SCADA interface of SIC as a real-time controller for a real canal, without needing to re-write the control algorithm. This is particularly interesting because it avoids possible errors and bugs in the process of implementing a controller on a real canal.

## Simulation Results

We give below an example of automatic tuning of distant downstream PI controller for a pool of Gignac Canal, using SIC software as simulator.

The considered pool is the feeder canal between Belbezet and Partiteur cross-regulators, which is 4 km long. The water level at Partiteur cross-regulator is measured every 5 minutes ( $T_s = 300$  s), and the gate opening at Belbezet cross-regulator is modified according to the ATV method, with a relay of  $d = 5$  cm. Fig. 11 shows the result of this simulation. After four oscillation cycles, the ATV control module of SIC measured the amplitude of the output  $a = 4.3$  cm, and derived the following ultimate cycle parameters:  $k_u = 2.96$  and  $T_u = 1440$  s.

We used these ultimate cycle parameters, to tune various PI controllers, which are automatically tuned with the robust rules (22–23), in order to get various gain margins between 6 and 14 dB, while keeping a constant phase margin ratio  $\Delta\Phi/\Delta\Phi_{\max} = 0.7$ .

This controller is automatically switched on after the last oscillation cycle, and the gate outlet at Partiteur is opened at time  $t = 15$  h, to simulate an unpredicted water withdrawal of  $0.07$  m<sup>3</sup>/s, which represents 10 % of the total discharge flowing in the canal  $0.70$  m<sup>3</sup>/s. The PI controllers react accordingly by opening the gate at Belbezet cross-regulator, to compensate for this decrease in the controlled water level. These simulations show the ability of the automatic tuning method to efficiently tune PI controllers for an irrigation canal pool.

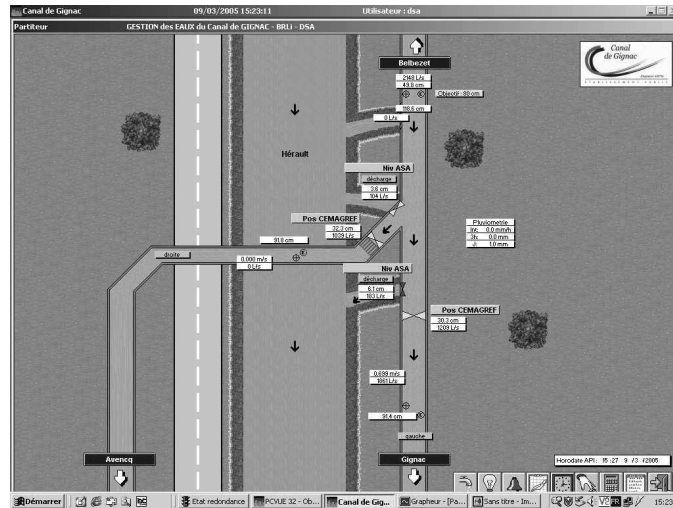


Figure 8: Interface of Gignac Canal SCADA supervisor for Partiteur location

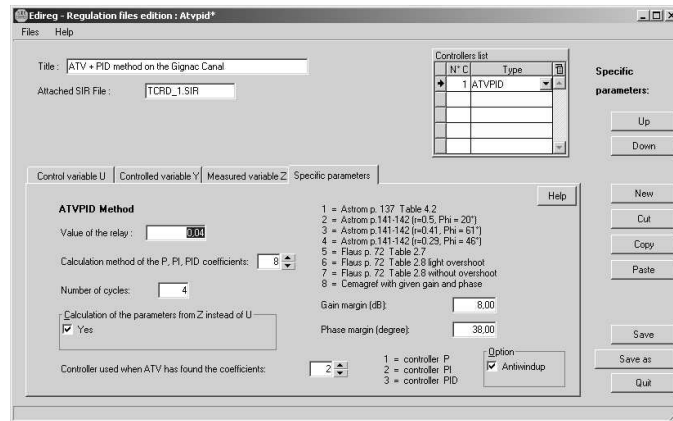


Figure 9: SIC interface for ATV method.

## Experimental Results

The purpose of the paper is not to derive the “best” tuning rule for a PI controller of an irrigation canal. The proposed automatic tuning method is tested experimentally to demonstrate its ability to identify pool dynamics in a single experiment, and to easily derive controller parameters, according to given tuning rules. Therefore, in the following experimental results, various gain margins will be tested according to the desired performance of the time response. A lower gain margin will lead to a more aggressive controller, which may be less robust to model variations, while a larger gain margin will lead to a more robust but slower controller. Similar rules are obtained for the phase margin. This issue of controller performance versus robustness margins has already been studied in previous papers (see Litrico and Fromion (2006) and Litrico et al. (2006)).

The paper is illustrated with three real-life experiments conducted on the Gignac Canal with the real-time SCADA interface of SIC software. The conditions of the three field experiments

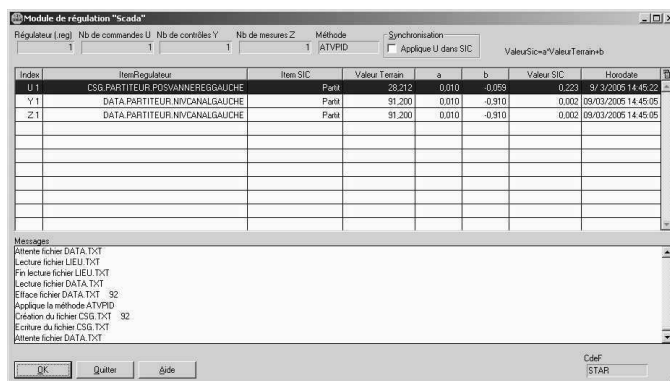


Figure 10: SIC interface for SCADA real-time control.

Table 3: Conditions of the field experiments on Gignac Canal

	control logic	goal of the test	flow rates	$T_s$
Exp. 1	local upstream	disturbance rejection	0.4–0.7 m <sup>3</sup> /s	120 s
Exp. 2	local downstream	set-point following	0.25–1.3 m <sup>3</sup> /s	60 s
Exp. 3	distant downstream	disturbance rejection	0.4 m <sup>3</sup> /s	300 s

are summarized in table 3, and are described in detail below.

### Exp. 1: Automatic Tuning of Local Upstream Controller

First, a local upstream controller is tuned with the ATV method on the Avencq cross regulator located at 4.5 km from the head gate on the right bank canal. This sluice gate is 1 meter wide. The water level upstream of the gate is measured every 2 minutes ( $T_s = 120$  s), and the gate opening is modified according to the ATV method, with a relay of  $d = 4$  cm. Fig. 12 shows the result of this experiment. After four oscillation cycles, the ATV module used by the SCADA interface of SIC measured the amplitude of the output  $a = 0.65$  cm, and derived the following ultimate cycle parameters:  $k_u = -7.8$  and  $T_u = 240$  s.

Based on these ultimate cycle parameters, a PI controller is automatically tuned with the robust rules (22–23), in order to get a gain margin of 10 dB and a phase margin of  $43^\circ$ . Therefore, using Eqs. (25–26), the controller parameters are obtained as  $k_p = -2.9$  and  $T_i = 360$  seconds. The PI controller is then automatically switched on, in order to control the upstream level around 79 cm. It does a good job when there is no perturbation. Then, an offtake located just upstream is opened at time  $t = 120$  minutes after the beginning of the test. The discharge in the canal is about 0.7 m<sup>3</sup>/s, and the offtake withdraws about 0.3 m<sup>3</sup>/s. The controller reacts rapidly, and rejects the perturbation in about 30 minutes.

At time  $t = 172$  minutes, the offtake is closed. The controller’s reaction is also good, and rejects the perturbation in about 40 minutes.

In this case, the automatic tuning method enables us to tune a robust controller in about 20 minutes, which is very quick. Since one cycle is 4 minutes (for the chosen sampling period of 2 minutes), a local upstream controller can be tuned with a few oscillation cycles, in our case 4 cycles, corresponding to a 16 minutes experiment.

This can be done by any canal manager without knowledge in automatic control, since all the

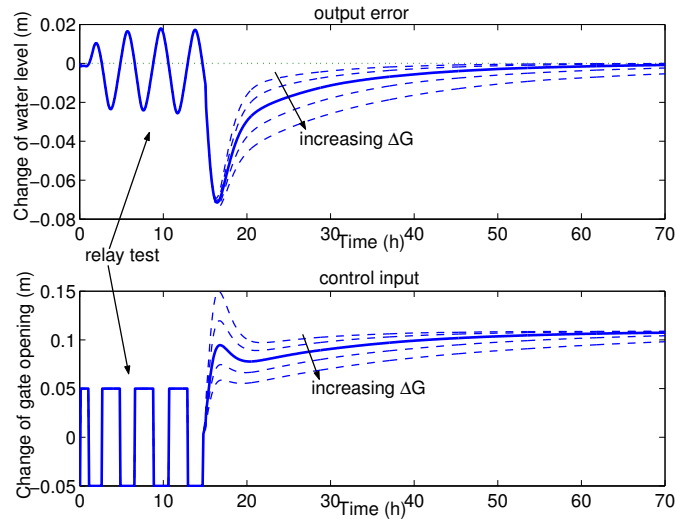


Figure 11: Simulation results for automatic tuning of a distant downstream PI controller between Belbezet cross-regulator and Partiteur cross-regulator, and perturbation rejection. Comparison of various gain margins  $\Delta G = [6, 8, 10, 12, 14]$  dB, and a constant phase margin ratio equal to 0.7. The solid line corresponds to  $\Delta G = 10$  dB.

procedure is automatic and programmed in the SIC software. It can be used either in simulation (using ATV control module of SIC) or on a real canal (using ATV control module of SIC called by the SCADA interface of SIC). To the best of our knowledge, SIC is the only software on the market which is able to do this, making it an efficient tool to transfer research advances towards the engineering world.

## Exp. 2: Automatic Tuning of Local Downstream Controller

The ATV method is used to tune a local downstream controller at the Partiteur Left Bank cross regulator. This sluice gate is 1.57 meter wide.

For this experiment, the downstream level of the gate is measured each minute ( $T_s = 60$  s), and the gate opening is manipulated according to the ATV method, with a relay of  $d = 4$  cm. Fig. 13 shows the result of this experiment. After four oscillation cycles, the amplitude of the output is  $a = 3.4$  cm, which gives the following ultimate cycle parameters:  $k_u = 1.5$  and  $T_u = 120$  s. Based on these ultimate cycle parameters, a PI controller is automatically tuned with the new robust tuning rules, with  $\Delta G = 9$  dB and  $\Delta\Phi = 30^\circ$ . Therefore, using Eqs. (22–23), the controller parameters are obtained as  $k_p = 0.60$  and  $T_i = 96$  seconds. The PI controller is then automatically switched on with these parameters, and a set-point following is tested. The level goes from 67 to 91 cm in 10 minutes, and then stabilizes. Another change in set-point is tested, to reach 96 cm. The controller does a good job, and can efficiently follow set-point changes. The initial flow rate is  $0.25$  m<sup>3</sup>/s, and the set-point following leads to a flow rate of  $1.3$  m<sup>3</sup>/s at the end of the experiment. The controller appears to be robust to large flow rate variations.

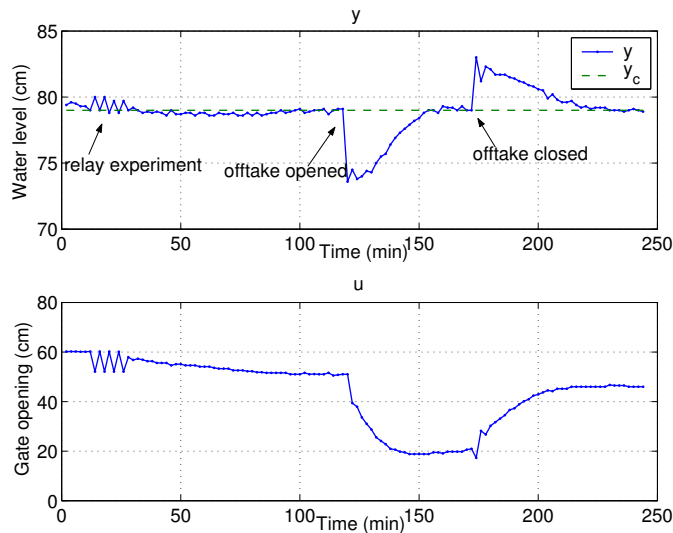


Figure 12: Automatic tuning of a local upstream controller at Avencq cross-regulator, and perturbation rejection (Exp. 1).

### Exp. 3: Automatic Tuning of Distant Downstream Controller

A distant downstream controller is tuned with ATV on one pool of the Gignac Canal between Lagarel and Mas Rouvière cross regulators (see Fig. 7). This pool is 4 km long, with an average slope of  $2.4 \times 10^{-4}$  m/m. The sluice gate at Lagarel cross regulator is 1 meter wide.

Fig. 14 shows the result of an ATV experiment to tune a distant downstream controller operating the gate at Lagarel to control the level at Mas Rouvière.

The upstream level of the gate is measured each 5 minutes ( $T_s = 300$  s), and the gate opening is modified according to the ATV method, with a relay of  $d = 5$  cm. For distant downstream control, since there may be some diffusion due to the transfer inside the pool it is necessary to choose a sufficiently large relay value so that its effect can be clearly seen at the downstream end of the pool.

After four oscillation cycles, the system measured the amplitude of the output  $a = 0.65$  cm, and derived the following ultimate cycle parameters:  $k_u = 9.79$  and  $T_u = 5400$  s.

Based on these ultimate cycle parameters, a PI controller is automatically tuned with the robust rules (22–23), in order to get a gain margin of 10 dB and a phase margin of  $43^\circ$ . Therefore, using Eqs. (25–26), the controller parameters are obtained as  $k_p = 3.62$  and  $T_i = 8140$  seconds. The PI controller is then automatically switched on, in order to control the water level at Mas Rouvière cross-regulator around 83.7 cm. The controller reacts rapidly to a drop in the downstream water level, by opening the upstream gate, then regulates the level close to the set-point.

This experiment shows the ability of the ATV method to tune a PI controller for distant downstream control of a real canal, in a limited time, without any prior knowledge of the system's dynamics.

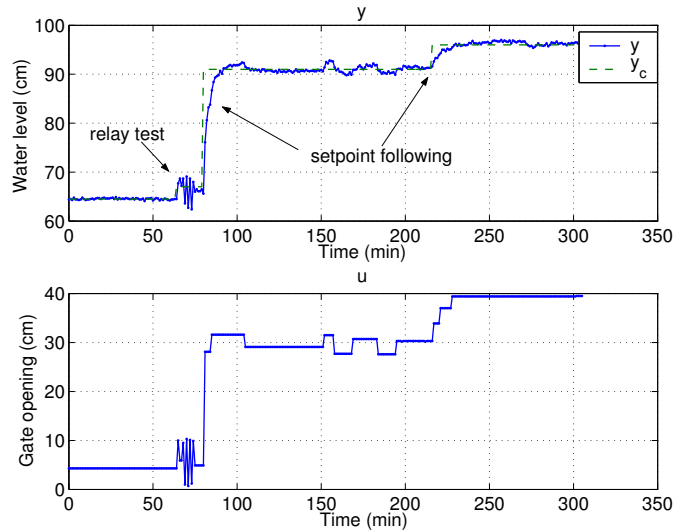


Figure 13: Automatic tuning of a local downstream controller at Partiteur cross-regulator, and set-point following (Exp. 2).

Table 4: Parameters obtained from ATV experiments on Gignac Canal

	$d$ (cm)	$a$ (cm)	$k_u$	$T_u$	$T_s$
Exp. 1	-4	0.65	-7.8	240	120
Exp. 2	4	3.40	1.5	120	60
Exp. 3	5	0.65	9.79	5400	300

## Discussion

### Comparison Between Experiments and Simulations

The experimental results are compared in this section to the one obtained by simulations using SIC software.

The experimental results are summarized in Table 4. The experiments numbers in the table correspond to the order of presentation of experimental results.

We built a complete model of Gignac Canal, which has been calibrated on data from the 2004 irrigation season. The three experiments have been reproduced in the model, to enable comparison between simulation and reality. Table 5 gives the parameters obtained from simulation on SIC software reproducing the experiments done on the Gignac Canal.

One observes that the ultimate period is correctly estimated using the simulator, which suggests that the Saint-Venant equations correctly represent the water flow dynamics. However, the

Table 5: Parameters obtained from ATV simulations on SIC model

	$d$ (cm)	$a$ (cm)	$k_u$	$T_u$	$T_s$
Exp. 1	-4	0.58	-8.83	240	120
Exp. 2	4	4.74	1.07	120	60
Exp. 3	5	0.98	6.49	5400	300

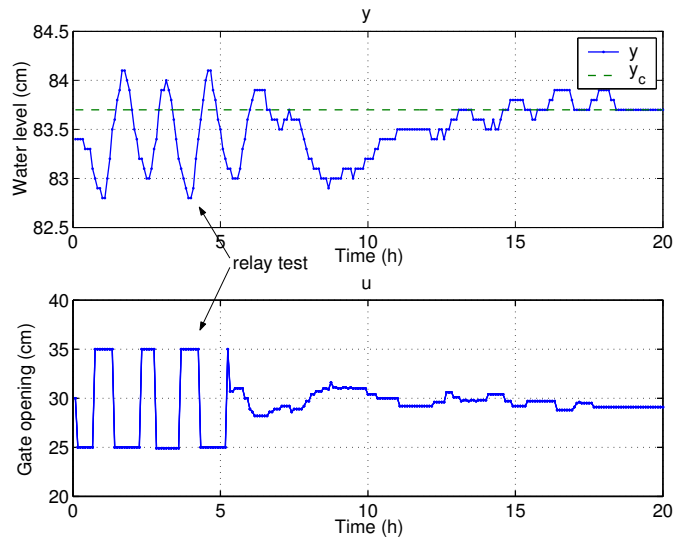


Figure 14: Automatic tuning of a distant downstream controller between Lagarel and Mas Rouvière cross-regulators (Exp. 3).

ultimate gain is not exactly recovered with the simulations. This is maybe due to an incorrect calibration of gates in the model. The discrepancy concerning the ultimate gain may also be due to real features not included in the model, to variations in the geometry of the canal, possibly due to weed growth, silt deposit, or to changes in the friction coefficient from the 2004 season. Nevertheless, the obtained coefficients are in the same order of magnitude as the ones obtained on the real canal. The ATV method is designed to be implemented on the real system, to take into account its real features, which are not always included in the model.

### Advantages and Drawbacks of the ATV Method

This section summarizes the advantages and drawbacks of the automatic tuning method. It has many interesting features:

- it provides robust P, PI or PID controller parameters based on a simple experiment,
- it naturally takes into account all the features of the system, including its neglected dynamics, communication delays, etc.
- it does not require a large measuring campaign,
- finally, as programmed in SIC software, it allows automatic controller tuning in simulation or on a real canal without any knowledge in automatic control.

However, the method also has some drawbacks:

- it does not take into account model variations, since it is based on the canal around a functioning point,
- since the control is the gate opening, the method does not directly give the value of the decouplers to choose in the case of successive multiple pools (Schuurmans et al., 1999b).

The first drawback can be overcome, since it is possible to do the experiment for different hydraulic conditions and then tune a robust controller using the obtained model parameters. Another possibility is to automatically tune the controller with ATV when the flow conditions have changed. We are working on a solution for the second drawback, in order to automatically tune decentralized controllers with decouplers for irrigation canals.

## Conclusion

The paper has presented and validated a method to automatically tune a P, PI or PID controller for an irrigation canal pool. The method is based on a relay experiment, which leads to small amplitude oscillations of the canal, in order to estimate the ultimate gain and ultimate frequency of the pool. The test signal is automatically generated by a relay inserted in the feedback loop and it can be used to automatically tune P, PI or PID controllers. This method does not require advanced automatic control knowledge and is implemented in SIC software, developed by Cemagref, which also incorporates a SCADA module for real-time control.

The extension of the method to multiple pools irrigation canals and the computation of the decoupler value is currently subject of research at Cemagref and will lead to future publications and implementation in the SIC software.

## Acknowledgments

The authors acknowledge the financial help of Région Languedoc-Roussillon, Conseil Général de l'Hérault, Agence de l'Eau Rhône-Méditerranée-Corse and Cemagref through the "Plateforme expérimentale du canal de Gignac".

We thank Vincent Kulesza, head of the ASA de Gignac, for his collaboration concerning the experiments.

## Appendix I. Ultimate Cycle Parameters Obtained via a Relay Experiment for an Integrator Delay Model

Let us compute the ultimate cycle parameters obtained via a relay experiment for an ID model given by equation (2). The system is assumed to be in steady state with persistent oscillations. We suppose without loss of generality that the error becomes negative at  $t = 0$ . Due to the integrator and since this output error comes from an input negative step of amplitude  $d$ , the error is decreasing as a negative ramp of slope  $-d/A_d$ . Then, at  $t = 0$  the relay leads to an input positive step of amplitude  $d$ . At  $t = \tau_d$ , this positive step influences the output, which has reached the value  $-d \times \tau_d/A_d$ . Then the output increases as a positive ramp of slope  $d/A_d$ , during a time equal to  $2\tau_d$ . This is depicted in Fig. 15.

Therefore, the amplitude of the output is equal to:

$$a = d \times \frac{\tau_d}{A_d}$$

and using equation (1), the ultimate parameters are given by:

$$k_u = \frac{4A_d}{\pi\tau_d}$$

and

$$T_u = 4\tau_d$$

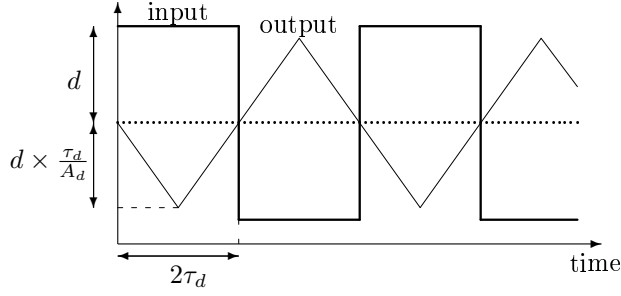


Figure 15: Relay experiment for an ID model

## Appendix II. Computation of Gain and Phase Margins

### Definition of Robustness Margins

Let us recall the definition of the considered robustness margins. The absolute gain margin  $\delta g$  is the maximum multiplicative increase in the gain of the system such that the closed-loop remains stable. The gain margin is denoted  $\Delta G$  when expressed in dB (i.e.,  $\Delta G = 20 \log_{10}(\delta g)$ ). The phase margin  $\Delta \Phi$  expressed in degrees is the maximum additive phase to the system such that the closed-loop remains stable.

We denote  $G_1(s)$  the considered process transfer function, and  $K_1(s)$  the considered controller transfer function. To study the frequency response of the controlled system, the open-loop transfer function  $K_1(s)G_1(s)$  is evaluated on the imaginary axis  $s = j\omega$ , where  $\omega$  is the frequency and  $j$  the imaginary number such that  $j^2 = -1$ . Two frequencies are of great interest when studying the open-loop of a controlled system: the crossover frequency where the gain of the open-loop equals 1, denoted  $\omega_c$ , and the frequency where the phase of the open-loop equals  $-180^\circ$ , denoted  $\omega_{180}$ . These two frequencies are critical ones, since the phase margin is evaluated for  $\omega = \omega_c$  and the gain margin is evaluated for  $\omega = \omega_{180}$ .

Then, to compute the gain and phase margins, one needs to first determine the frequencies  $\omega_c$  and  $\omega_{180}$ , by solving the equations:

$$|K_1(j\omega_c)G_1(j\omega_c)| = 1 \quad (32)$$

$$\arg(K_1(j\omega_{180})G_1(j\omega_{180})) = -180 \quad (33)$$

$\Delta G$  and  $\Delta \Phi$  are then determined from the equations:

$$|K_1(j\omega_{180})G_1(j\omega_{180})| = 10^{-\frac{\Delta G}{20}} \quad (34)$$

$$\arg(K_1(j\omega_c)G_1(j\omega_c)) = -180 + \Delta \Phi \quad (35)$$

This can be done numerically, using the command `margin` of Matlab software.

### Computation of Gain and Phase Margins of a PI Controller for a ID Model

Let us now consider the case of a canal pool controlled by a PI controller. The ID model is given by

$$G_1(s) = \frac{e^{-\tau_d s}}{A_d s} \quad (36)$$

with  $\tau_d$  the propagation delay of the pool (s) and  $A_d$  the inverse integrator gain. The PI controller is denoted by:

$$K_1(s) = k_p \left( 1 + \frac{1}{T_i s} \right) \quad (37)$$

with  $k_p$  the proportional gain and  $T_i$  the integral time.

This leads to:

$$|K_1(j\omega)G_1(j\omega)| = \frac{k_p}{A_d \omega} \sqrt{1 + \frac{1}{T_i^2 \omega^2}} \quad (38)$$

$$\arg(K_1(j\omega)G_1(j\omega)) = -180 - \frac{180}{\pi} \tau_d \omega + \frac{180}{\pi} \arctan(T_i \omega). \quad (39)$$

The frequency  $\omega_{180}$  can be easily approximated if we assume that  $\arctan(T_i \omega_{180}) \approx \pi/(2\tau_d)$ . Such an approximation is valid if  $1/T_i \ll \omega_{180}$ , typically if  $T_i \gg \tau_d$ . Then, using Eq. (39), one gets:

$$\omega_{180} = \frac{\pi}{2\tau_d}. \quad (40)$$

Solving Eq. (32) leads to the following expression for the crossover frequency  $\omega_c$

$$\omega_c = \frac{k_p}{\sqrt{2}A_d} \sqrt{1 + \sqrt{1 + \frac{4A_d^2}{T_i^2 k_p^2}}} \quad (41)$$

Using the approximate relation between the crossover frequency  $\omega_c$  and  $\omega_{180}$ , given by:

$$\omega_c = \omega_{180} 10^{-\frac{\Delta G}{20}}, \quad (42)$$

one obtains the gain margin as a function of  $\omega_c$  and  $\omega_{180}$ :

$$\Delta G = 20 \log_{10} \left( \frac{\omega_{180}}{\omega_c} \right). \quad (43)$$

Using the definition of the phase margin  $\Delta\Phi$ , the phase of the open-loop at the crossover frequency  $\omega_c$  is given by:

$$\arg(K_1(j\omega_c)G_1(j\omega_c)) = -180 + \Delta\Phi.$$

Combining with Eq. (39), this leads to:

$$\Delta\Phi = \frac{180}{\pi} (\arctan(T_i \omega_c) - \tau_d \omega_c). \quad (44)$$

## Notations

The following notations are used in this paper:

$A_d$  = inverse integrator gain;

$a$  = amplitude of the measured output in m;

$d$  = amplitude of the relay in m;

$e$  = output error in m;

$G_1(s)$  = system transfer function;

$j$  = complex number  $j^2 = -1$ ;  
 $K_1(s)$  = controller transfer function;  
 $k$  = variable gain;  
 $k_p$  = proportional gain;  
 $k_u$  = ultimate gain;  
 $n$  = filter parameter for PID controller;  
 $N$  = number of oscillation cycles;  
 $r$  = set-point for controlled variable;  
 $s$  = Laplace variable in  $s^{-1}$ ;  
 $T_d$  = derivative time in s;  
 $T_i$  = integral time in s;  
 $T_s$  = sampling time period in s;  
 $T_u$  = ultimate period in s;  
 $t$  = time in s;  
 $u$  = control action variable (gate opening in m);  
 $y$  = controlled variable (water elevation in m);  
 $\delta g$  = absolute gain margin;  
 $\Delta G$  = gain margin in dB;  
 $\Delta \Phi$  = phase margin in degrees;  
 $\Delta \Phi_{\max}$  = maximum phase margin in degrees;  
 $\omega$  = frequency in rad/s;  
 $\omega_{180}$  = frequency in rad/s;  
 $\omega_c$  = crossover frequency in rad/s;  
 $\tau_d$  = time-delay for downstream propagation in s;

## References

- Aström, K. and Hägglung, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5):645–651.
- Aström, K. J. and Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, Research Triangle Park, NC, second edition. 351 p.
- Baume, J.-P., Malaterre, P.-O., and Sau, J. (1999). Tuning of PI to control an irrigation canal using optimization tools. In *Workshop on Modernization of Irrigation Water Delivery Systems*, pages 483–500, Phoenix, Arizona, USA.
- Chien, Hrones, and Reswick (1952). On the automatic tuning of generalized passive systems. *Trans. ASME*, (74):175–185.
- Clemmens, A. J. and Schuurmans, J. (2004). Simple optimal downstream feedback canal controllers: theory. *J. Irrig. Drain. Engrg.*, 130(1):26–34.
- Hang, C., Astrom, K., and Wang, Q. (2002). Relay feedback auto-tuning of process controllers – a tutorial review. *Journal of Process Control*, 12:143–162.
- Litrice, X. and Fromion, V. (2004). Simplified modelling of irrigation canals for controller design. *J. Irrig. Drain. Engrg.*, 130(5):373–383.

- Litrice, X. and Fromion, V. (2006). Tuning of robust distant downstream PI controllers for an irrigation canal pool: (I) Theory. *J. Irrig. Drain. Engrg.* 132(4): in press.
- Litrice, X., Fromion, V., and Baume, J. (2006). Tuning of robust distant downstream PI controllers for an irrigation canal pool: (II) Implementation issues. *J. Irrig. Drain. Engrg.* 132(4): in press.
- Malaterre, P.-O. (2006). SIC 4.22, Simulation of Irrigation Canals. <http://www.cemagref.net/sic/sicgb.htm>.
- Malaterre, P.-O. and Baume, J.-P. (1997). SIC 3.0, a simulation model for canal automation design. In *Int. Workshop on the Regulation of Irrigation Canals RIC'97*, Marrakech (Morocco).
- Malaterre, P.-O., Rogers, D. C., and Schuurmans, J. (1998). Classification of canal control algorithms. *J. Irrig. Drain. Engrg.*, 124(1):3–10.
- Schuurmans, J., Clemmens, A. J., Dijkstra, S., Hof, A., and Brouwer, R. (1999a). Modeling of irrigation and drainage canals for controller design. *J. Irrig. Drain. Engrg.*, 125(6):338–344.
- Schuurmans, J., Hof, A., Dijkstra, S., Bosgra, O. H., and Brouwer, R. (1999b). Simple water level controller for irrigation and drainage canals. *J. Irrig. Drain. Engrg.*, 125(4):189–195.
- Tyreus, B. and Luyben, W. (1992). Tuning PI controllers for integrator/dead time processes. *Ind. Engrg. Chem. Res.*, (31):2625–2628.
- Weyer, E. (2001). System identification of an open water channel. *Control Engineering Practice*, 9:1289–1299.
- Weyer, E. (2002). Decentralised PI controller of an open water channel. In *15th IFAC World Congress*, Barcelona, Spain.
- Ziegler, J. and Nichols, N. (1942). Optimum settings for automatic controllers. *Trans. ASME*, (64):759–768.