GoRoSoBo simplified: an accurate feedback control algorithm in real time for irrigation canals

Enrique Bonet, Manuel Gómez, M. T. Yubero and J. Fernández-Francos

ABSTRACT

An irrigation canal is a hydraulic system whose main objective is to convey water from a source (dam, river) to different users. Such systems can be very large (several tens or hundreds of kilometers), characterized by time delays and non-linear dynamics, strong unknown perturbations and interactions among subsystems. In order to fulfill the requirements of canal users, the water manager must control all water deliveries during the irrigation cycle (or irrigation program) calculating the gate positions of the canal according to the water demands in real time. Initially, our overall control diagram in real time is mainly represented by two algorithms, the canal survey estimation algorithm (this algorithm estimates the water level and velocity along the irrigation canal during a past time horizon) and GoRoSoBo algorithm (feedback control algorithm operating in real time). Regarding long canals with several gates and pumps operating in a short period of time for a long predictive horizon, the initial version of GoRoSoBo algorithms would spend too much time calculating the canal gate position in real time. This is the reason why we have upgraded the code of the GoRoSoBo algorithm, saving in computational time around 85%, in order to operate in real time.

Key words | agricultural demands, canal control, flow rate extractions, open channel flow, optimization algorithms

Enrique Bonet (corresponding author)
Manuel Gómez
M. T. Yubero
Barcelona School of Civil Engineering,
Flumen Institute,
C/Jordi Girona, 1, 08034, Barcelona,
Spain
E-mail: *enrique.bonet@upc.edu*

J. Fernández-Francos
Hydraulic Engineering Section, Energy Department,
Polytechnic School of Mieres (University of Oviedo),
C/Gonzalo Gutiérrez Quirós s/n. 33600-Mieres, Asturias,
Spain

ACRONYMS

AEMET	State Agency of Spanish Meteorology
ASCE	American Society of Civil Engineers
ATV-PID	Auto Tuning Variation – Proportional Integral Derivative control
BIVAL	Constant downstream volume system
CACG	Compagnie d'Aménagement des Coteaux de Gascogne
CARA	Compagnie d'Aménagement Rural d'Aquitaine
CARDD	Canal Automation for Rapid Demand Deliveries
CEMAGREF	Centre National du Machinisme Agricole, du Génie Rural, des Eaux et des Forêts
CFL	Courant Friedrichs Levi
CLIS	Constant level control method based on the inverse solution of the Saint-Venant equations
CNABRL	Compagnie Nationale d'Aménagement du Bas-Rhône Languedoc
CPU	Central processing unit
CSE	Canal survey estimation
CUDA	Compute Unified Device Architecture
HIM	Hydraulic influence matrix
GoRoSo/GRS	Gómez, Rodellar, and Soler (feed-forward algorithm)
GoRoSoBo/GRSB	Gómez, Rodellar, Soler, and Bonet (feedback algorithm)
GPC	Generalized predictive control method
IAE	Integral of absolute magnitude of error

doi: 10.2166/hydro.2019.159

IAQ	Integrated absolute discharge change
ICT	Information and communications technology
ID	Integrator delay
INE	Instituto Nacional de Estadística (Statistics National Institute)
LQR	Linear quadratic regulator
MAE	Maximum absolute error
MCG	Modelos de circulación general
MCR	Modelos climáticos regionales
MIMO	Multiple-input multiple-output
MISO	Single-input single-output
MMA/MIMAM	Ministerio de Medio Ambiente (Ministry of Environment)
PILOTE	Preissmann Implicit scheme, Linear Optimal control, Tracking of variables and Estimation of perturbations
PI	Proportional integral
PID	Proportional integral derivative

INTRODUCTION

One of the most important problems for an on-demand irrigation system designer is the calculation of the discharges flowing into the network. Such discharges strongly vary over time depending on the cropping pattern, meteorological conditions, on-farm irrigation efficiency, and farmers' behavior. The amount of water supplied to the farmers during an irrigation event, referred to as the target or required depth of application, is the major design consideration. The unknown changes in water demands are quite usual in a canal during an irrigation cycle; this is the reason why the real extracted flow is unknown by the water manager. In that case, it is quite difficult for any control algorithm calculating new gate trajectories to satisfy the general scheduled deliveries when the real extracted flow is unknown as well as the current canal state. In a hypothetical case where the water manager knows the real water demand in the canal in real time, the feedback control algorithm should also calculate the new gate trajectories as soon as possible in order to make optimum canal control.

On the other hand, not all problems in canal management are due to water delivery operations but problems associated with canal lining. In a particular case where fast gate movements are proposed by the feedback control algorithm, due to increasing water demands, it could provoke a dangerous event for the canal. In this case, significant flow changes (operating conditions) are introduced in the canal to fulfill water demand which could cause overflow or dry the canal and in extreme conditions (high groundwater level) could crack the canal lining. The canal overflow is produced when the water level exceeds the top of the canal bank, caused by the transient wave. During extended operations, hydrostatic pressure develops on both sides of the canal lining as the supporting soils become saturated. When the canal water is removed, the hydrostatic pressure behind the lining is no longer offset by the water in the canal. The differential hydrostatic pressure can cause the canal lining to buckle outwards or separate from adjacent linings (Figure 1).

In that sense, there are several reasons for enhancing strong changes in the canal state which implies increasing or decreasing on gate position to control the water level according to unscheduled water demands. This is the reason why some authors make their control algorithms more in terms of quicker responses than accuracy, although many of these algorithms are always within high accuracy standards.

There are several control algorithms for irrigation canals with high performance such as the following:

- CLIS (Liu *et al.* 1998), based on an inverse solution method of the Saint-Venant equations and designed for the automation of demand-oriented systems.

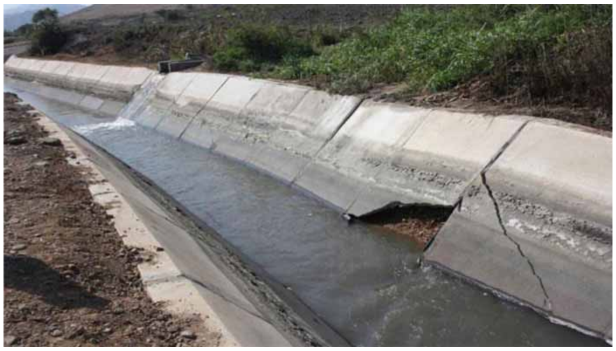

Figure 1 | Cracking of a cross section of a canal due to differential hydrostatic pressure (Chimbote canal (Peru)).

- The PILOTE (Malaterre 1995) is a LQR closed-loop controller and obtained from the steady-state solution of the Riccati equation. A Kalman filter is used to reconstruct the state variables and the unknown perturbations from a reduced number of observed variables.
- A distributed model predictive controller or DMPC (Álvarez *et al.* 2013) is an algorithm which initially solves several local MPC problems obtaining cost function and sharing this information among controllers to consider the control sequence for the neighbor controller that gets the smallest value of its own cost function.
- A predictive control (Aguilar *et al.* 2016) which generates gate openings from predictive control law with fixed parameters regarding a simplified dynamic model.

The first two control algorithms use in their formulations the first derivative of gate position with respect to the discharge or the water level for an initial steady state. These kinds of simplifications provide to the calculation process a reduction of computation time but sometimes less accuracy of the result depending on the changes in canal state. All these shortcomings make quite difficult managing an irrigation canal and the reason to propose our simplified overall control diagram. Instead, the DMPC or the predictive controller are based on a process model that is used to predict the evolution of the system state, that is, the water level and velocity output along a prediction horizon. This process model is usually a simplification of the full Saint-Venant equations and frequently are linear models.

The main objective is to develop a simplified overall control diagram able to operate the canal in real time on large systems, rectifying the gate trajectory in case of disturbances, and re-establishing the desired behavior of the canal by the water manager. This simplified overall control diagram is tested numerically but it has not been evaluated in a real canal due to the difficulties of testing these kinds of algorithms in a real environment. This is the reason why Clemmens developed the test cases for canal control algorithms in 1998 (Clemmens *et al.* 1998). Algorithms have not been provided in this paper due to space limitations and other considerations.

The feedback control algorithm included in the overall control diagram is GoRoSoBo simplified. The initial version of GoRoSoBo updated the first derivative of the gate position in respect to water level in each time interval calculating gate trajectories with accuracy but increasing the computation time. In such cases, we propose to reduce the computation time of GoRoSoBo reformulating the control algorithm code, so the new control algorithm obtained is called GoRoSoBo simplified.

This paper is divided into several sections. The methodology applied in our overall control diagram is introduced in the next section as well as the HIM matrix and optimization problem. In the results and discussion sections, we evaluate GoRoSoBo and GoRoSoBo simplified in a simple canal test analyzing gate trajectory and water level at the checkpoint in both cases. The control diagram was initially tested in a canal with a single pool for a better understanding of the results obtained by GoRoSoBo and GoRoSoBo simplified; this geometry of the canal is based on Bautista's work (Bautista *et al.* 1997). This kind of canal has been proposed by several authors testing their control algorithms (Wylie 1969; Chevereau 1991; Liu *et al.* 1992; Soler *et al.* 2008; Bonet 2015). Although the main purpose of this section is to check the results obtained by GoRoSoBo and GoRoSoBo simplified on the ASCE test cases for canal 1 and 2 (Clemmens *et al.* 1998) introduced by the ASCE (Task Committee on Canal Automation Algorithms). In the performance indicator section, the results obtained by several controllers (including GoRoSoBo and GoRoSoBo simplified) are evaluated applying the performance indicators proposed by the ASCE Task Committee. The results in these performance indicators have been compared in order to check the degree of accuracy of our control algorithms.

METHODOLOGY

There are different ways of approaching an overall scheme for canal control, and we show our ideal solution of an overall control diagram of a canal in Figure 2. The on-line computation is the block focused on in this paper although the off-line computation block is also part of our overall scheme for canal control. In the case of the off-line component, the desired water levels in every outflow orifice are determined according to crop necessities during the

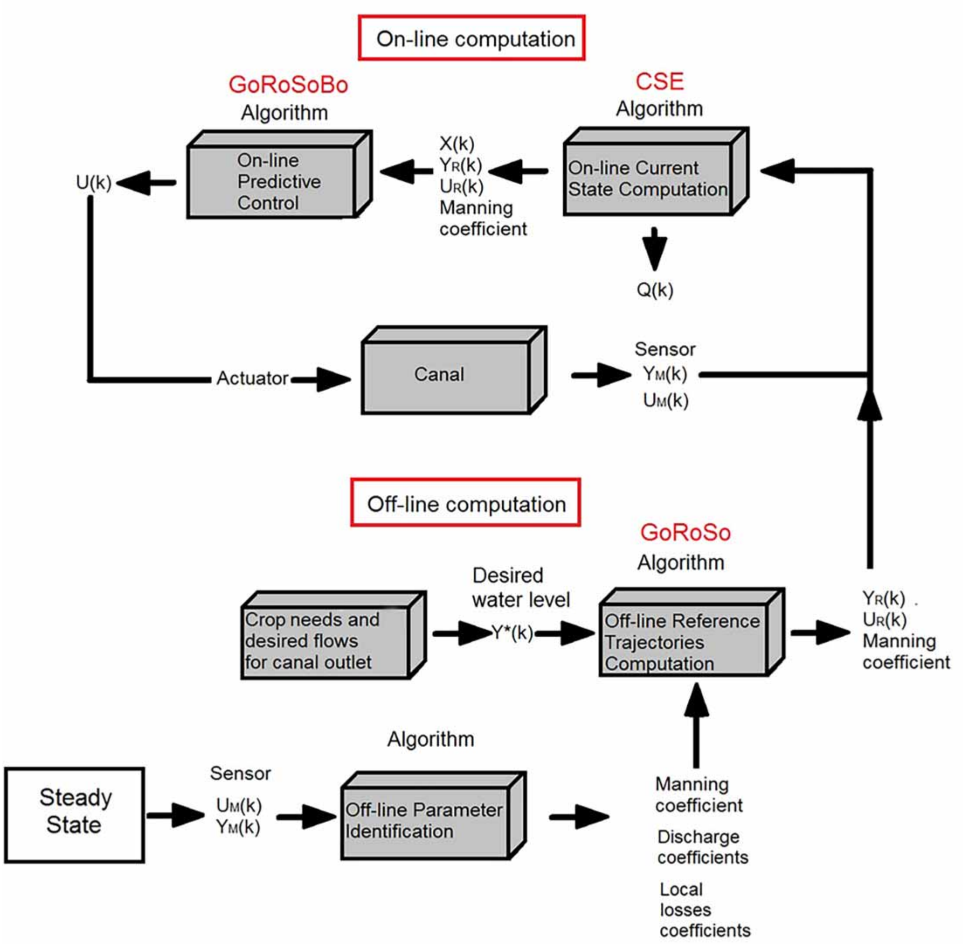

Figure 2 | Overall control diagram simplified of the irrigation canal.

irrigation cycle and the feedforward control algorithm (GoRoSo) calculates the optimum gate trajectories regarding the desired water levels; all this information will be used in the on-line computation block. In order not to introduce old aspects involved in off-line computation, we only focus on the on-line computation block in advance. Regarding the on-line computation block, CSE (canal survey estimation) and GoRoSoBo simplified algorithms are involved in this task and detailed in the next paragraphs.

1. 'Crop needs and desired hydrographs for canal outlets': The water demand is calculated according to the crop needs. Thus, the water demand is transformed to 'desired water level' (*Y**) at several cross sections.
2. 'Off-line computation of the reference trajectories': The desired water levels (*Y**) are provided to THE GoRoSo algorithm (off-line reference trajectory computation) which calculates the optimum gates trajectories (*Ur*) in the canal for the next irrigation cycle, according to Soler *et al.* (2008).
3. 'Off-line parameter identification': Hydraulics coefficients' estimation (Manning coefficient or gate discharge coefficient).
4. 'On-line current state somputation': This module uses all values stored in the database as '*U*' (gate position) and Y_M (measured water level) which have been measured at the checkpoints in the canal (in this paper the measured water levels were simulated by a model, we do not use measured water level from a real canal). If the water level measured at checkpoints is different from the desired water level, it is probably due to a disturbance introduced into the system. In those

circumstances, the CSE is an optimization algorithm (Bonet 2015) able to estimate the flow disturbances in the canal (that is, the real water demand in the canal) from the water level changes, among the measured water level and the desired water level at checkpoints. The algorithm can also obtain the current hydrodynamic state of the canal which is very useful in any case. The hydrodynamic state of a canal is defined as the velocity and water level at each cross section of the canal.

5. 'On-line predictive control': The reference gate trajectories, the disturbances, and the hydrodynamic canal state are provided to the control algorithm, called 'On-line predictive control', which must react on-line in case of water level changes among the measured water level and the desired water level at the checkpoints during the predictive horizon. The predictive control recalculates new gate positions (U) to come back to the reference behavior for a predictive horizon in every operation period. There is an extensive literature on feedback control algorithms which recalculate the gate trajectories as CARA (Marzouki 1989; Clemmens & Wahlin 2004; Wahlin & Clemmens 2006).

In this paper, we focus on the on-line control portion and really on the simplified approach because we are interested in improving the efficiency in water transport in 'real time'.

The development of CSE and GoRoSoBo algorithm is beyond the scope of this paper but can be viewed in Bonet (2015) or Bonet *et al.* (2017). GoRoSoBo and GoRoSoBo simplified algorithms use the HIM matrix (hydraulic influence matrix) to calculate the gate trajectories (1) (see Bonet 2015). The HIM matrix represents the impact of gate movements on the canal state (water level and velocity at every cross section of the canal) during the irrigation cycle which coincides with the testing period. Solving this system of Equation (1), the GoRoSoBo simplified algorithm can obtain the gate trajectories to apply in the irrigation canal considering the water level changes.

$$\begin{aligned} &[HIM'(U)]_0 = \frac{\partial Y_0}{\partial U_0} \\ &\Delta U = [HIM'(U)]_0^{-1} \Delta Y \end{aligned} \tag{1}$$

where ΔY represents the water level changes at selected cross sections of the canal, ΔU represents gate position changes, $HIM'(U)_0$ (Jacobian matrix) is the simplified hydraulic influence matrix that represents the impact of gate movements on the water level at different cross sections of the canal at a specific time step (before the irrigation cycle started).

The HIM matrix

In order to simulate the water surface profile on a canal, we consider the Saint-Venant equations developed in this section. This system of equations is a set of hyperbolic partial differential equations derived from equations of conservation of mass and conservation of linear momentum. These equations cannot be solved analytically so numerical approaches should be used to solve them. Several numerical schemes can be used such as finite difference (implicit or explicit), characteristics curves, and finite volume.

Like any hyperbolic system, it can be transformed into its characteristic form (characteristic curves). Such transformation of the Saint-Venant equations gives an ordinary system of four equations (Bonet 2015).

On the other hand, there are many potential control structures in canals. The individual study of each one is beyond the scope of this paper, so this is the reason why we introduce the most usual structure. The most usual canal structure is a checkpoint (Figure 3), a target point where the water level is measured with a depth gage, and it includes a sluice-gate, a lateral weir outlet, offtake orifice or a pump (shown in Bonet 2015; Bonet *et al.* 2016). The interaction of this control structure with the flow can be described according to the mass and energy conservation Equation (2).

$$\begin{aligned} S(y_e)\frac{dy_e}{dt} &= A(y_e)v_e - q_b - q_s(y_e) - A(y_s)v_s - q_{offtake}(y_e) \\ A(y_s)v_s &= k_c u\sqrt{y_e - y_s + d} \\ q_s(y_e) &= C_S a_S (y_e - y_0)^{3/2} \\ q_{offtake}(y_e) &= C_0 A_0 \sqrt{2g y_e} \\ k_c &= \sqrt{2g C_d a_c} \end{aligned} \tag{2}$$

- v_e is the weighted average velocity of all the particles in a canal cross section.
- y_e is the water level of all the particles in a canal cross section.

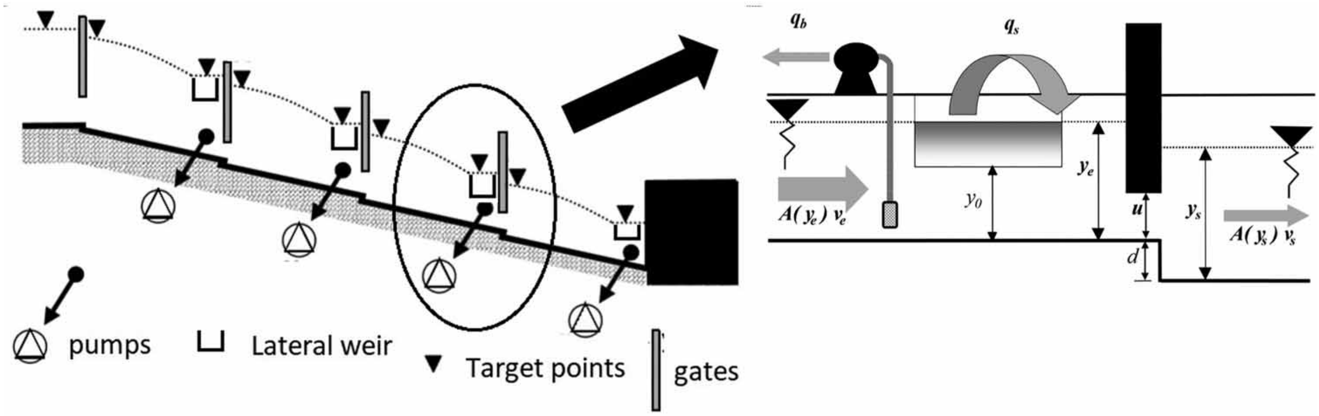

Figure 3 | Example of a canal profile and diagram of a checkpoint with gate, lateral weir, and pump.

- $S(y_e)$ is the horizontal surface of the reception area in the checkpoint.
- $A(y_e)^*v_e$ is the incoming flow to checkpoint, defined in terms of water level and velocity.
- $A(y_s)^*v_s$ is the outflow from the checkpoint which continues along the canal, described in terms of water level and velocity.
- C_d is the discharge coefficient of the sluice-gate and ac is the sluice-gate width.
- d is the checkpoint drop, and u is the gate opening.
- q_b is the pumping offtake.
- $q_s(y_e)$ is the outgoing lateral flow through the weir where C_s is the discharge coefficient, a_s is the weir width and y_0 is the weir height measured from the bottom, called weir equation.
- $Q_{offtake}(y_e)$ is the outflow orifice flow where C_0 is the discharge coefficient, A_0 is the area of the offtake orifice, called orifice offtake equation.

The presence of checkpoints or control structures in a canal lead to a sub-division into canal pools, that is, there is a canal pool between two checkpoints, and there is a checkpoint between two pools (Supplemental information, Figure S1).

The system of Saint-Venant and checkpoint equations cannot be solved analytically, so the use of numerical techniques is mandatory. In order to reduce integration time with a minimum loss of accuracy, we have adopted a second order finite differences discretization, particularly 'the method of characteristic curves' (Gómez 1988) (see Supplemental information section).

The system of Saint-Venant equations is solved numerically for every cross section at every time step (including checkpoint equations in particular cross sections). This process demands a huge computation time, although less demanding than calculating the Jacobian matrix from Saint-Venant equations which is mandatory to build the predicted model (HIM matrix). The HIM matrix in complete form defines the impact of any gate movement over the hydraulic behavior of the canal (that is, water level and velocity in any cross section of the canal), usually divided by pools from checkpoints. The HIM matrix calculation is a highly time-consuming process, especially when the canal is too long and there are many checkpoints, this can be a big problem working in real time. In that sense, the water manager does not have much time for canal operations in real time. Thus, the water manager must introduce the correct gate trajectories to keep the target levels considering the disturbances in a short period of time, so the control algorithm must also assist the water manager in a short period of time. This is the reason why we propose using the HIM and GRSB simplified. This control algorithm does not recalculate the HIM matrix between sampling intervals as GRSB. GRSB simplified calculates the HIM matrix in a single period of time (steady state), that is, the HIM simplified ($HIM'(U)_0$) and it uses this HIM simplified matrix during all irrigation cycles.

The optimization problem

To introduce the optimization problem, we have to evaluate some vectors included in the problem development. GRSB simplified algorithm needs as input data, the water level target at some points (checkpoints) for a predictive horizon established by the water manager. As we have noted before, the desired water level vector contains the water level targets at the checkpoints from the time instant 1 to k_F (3) and vector dimension is n_y, where $n_y = k_F \times n_c$, k_F is the final instant of the future time horizon and $\mathrm{n_c}$ is the number of checkpoints. We define this vector as:

$$Y^* = [y^*(1), y^*(2), \ldots, y^*(k_F - 1), y^*(k_F)]^T \tag{3}$$

We can check the desired water level vector values in a computational grid in Figure 4 (dots).

In another way, we can obtain 'prediction output vector' that contains the water level at several cross sections or checkpoints (n_c) from the time instant 1 to k_F. This vector is calculated from the output data of the CSE algorithm at the time instant k of all the discretization points.

$$y(k) = [y_1(k), \ldots, y_i(k), \ldots, y_{n_C}(k)]^T \tag{4}$$

The dimension of the prediction output vector is $n_Y = k_F \times n_c$. Vector (4) contains the water level in cross sections (checkpoints) introduced in Figure 4. On the other hand, the prediction output vector (4) is obtained at the same cross sections as the desired water level vector (3), as you can check in the grid domain at Figure 4.

As we showed before, GoRoSoBo calculates the gate trajectories for a predictive future horizon. In that case, the gates are moving for an operation period K (in all cases, the sampling period and operation period will be the same). Then, the gate trajectories can be approached with piecewise functions. The gate trajectories vector is defined by lumping together all the gate trajectories during the future horizon, as follows:

$$U = \left[U_1(1), \ldots, U_{n_g}(1), \ldots, \ldots, U_1(K_F), \ldots, U_{n_g}(K_F)\right]^T \tag{5}$$

where the dimension of this vector (gate trajectories) is $n_U = n_u \times K_F$, n_u is the number of gates and K_F is the final operation period of the future horizon.

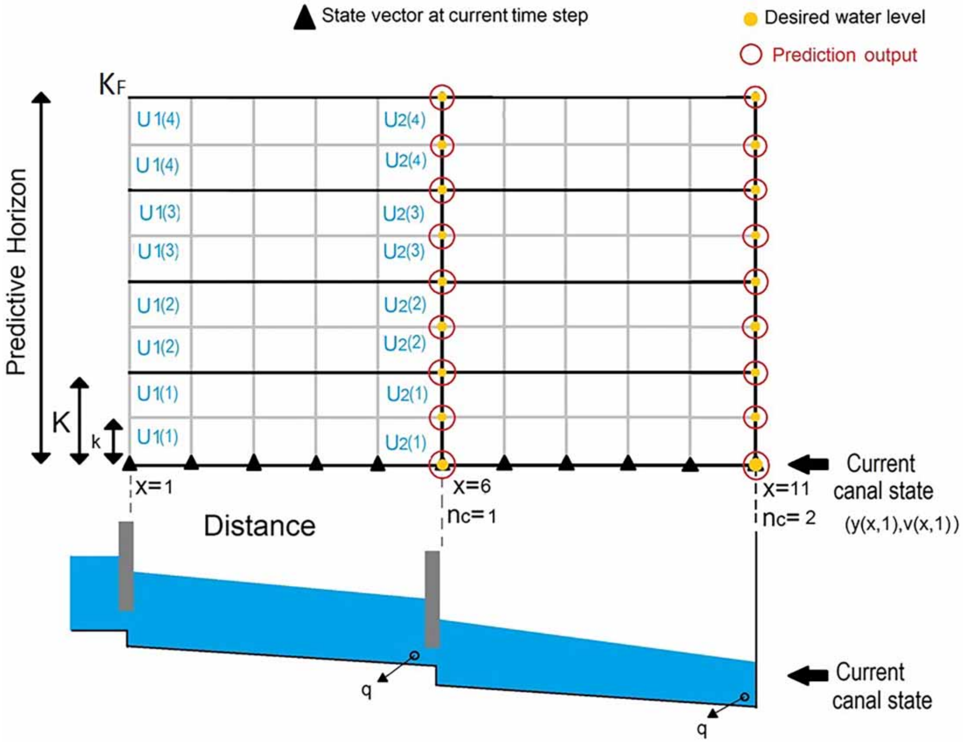

Figure 4 | Sketch of a numerical grid of a canal with two pools controlled by two checkpoints downstream of each pool. There are pump stations close to each checkpoint. Pump flow trajectories are defined with four operation periods. Also, it shows the *x*/*t*-dots where the flow behavior is defined. Notice that '*K*' with a capital letter denotes time interval of control during the future horizon and '*k*' with a small letter denotes time instant of simulation.

Therefore, the GoRoSoBo algorithm calculates the gate trajectories (ΔU) in order to come back to the desired water level (Y^*) regarding several measured water levels (that is, the prediction output vector) (ΔY) (1).

If we focus on the optimization problem, the objective is to make the prediction output vector more similar to the desired water level vector by manipulating the gate trajectories' vector (see Gill *et al.* 1981; Fletcher 1987). In mathematical terms, the objective is to obtain the gate trajectories' vector that minimizes the following performance criterion:

$$\begin{aligned} &Minimize\ J(U) = \frac{1}{2}\left[[C]X_{K_i}^{K_F}(U) - Y^*\right]^T [Q]\left[[C]X_{K_i}^{K_F}(U) - Y^*\right] \\ &r_k(U) = 0, i \in I(U) \\ &r_k(U) \geq 0, i \in NI(U) \qquad (6) \end{aligned}$$

where $J(U)$ is the objective function, $X_{ki}^{kf}(U)$ is the prediction vector from the operation time step K_i to K_F; Y^* is the desired water level vector; Q is a weighting matrix (identity-diagonal matrix); C is the discrete observer matrix, see Malaterre (1994); and $r_k(U)$ is the '*kth*' constraint function; $I(U)$ is a set of equalities constraints and $NI(U)$ is a set of inequalities constraints. U contains the gates' trajectories (5).

The optimization problem is solved using the SQP method which is a robust and applicable method for constrained optimization problems (Bonet 2015). This method uses the Hessian and HIM matrixes which are ill-conditioned matrixes, so Marquardt coefficient is introduced in the diagonal of the matrix improving its condition number and solving convergence problems of the method.

RESULTS AND DISCUSSION

Numerical example: a canal introducing a single disturbance

In this example, we test both control algorithms, GoRoSoBo and GoRoSoBo simplified, in a single canal and we analyze the results in both cases as well as the computation time used.

The geometry of the canal is based on Bautista *et al.* (1997). This canal was used by different authors such as Wylie (1969), Liu *et al.* (1992), Chevereau (1991), and Soler *et al.* (2008). Instead, boundary conditions of the test are based on Liu's example (scheduled demand).

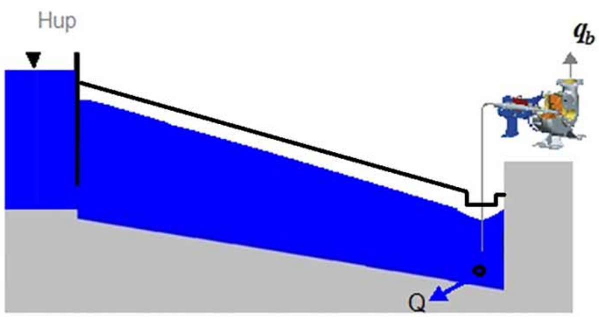

Figure 5 | Canal profile with a single pool.

The canal with a trapezoidal section is represented in Figure 5, and the general data (Manning's coefficient, canal depth, pool length, bottom slope, side slopes, and bottom width) are shown in Table 1. The characteristics of checkpoint, sluice gate, pump station and orifice offtake are shown in Table 2.

In this test an upstream large reservoir is considered, whose water level H_{up} is 3 m constantly throughout the test. At the end of the last pool, there is a control structure with an orifice offtake and a pump station. The flow through the orifice depends on the water level over the orifice, and the disturbance (unscheduled demand) is introduced by the pump station. This example starts from an initial steady state, where the upstream boundary condition is a flow rate through gate 1 of 5 m^3/s and the downstream boundary condition (at the end of the canal) is a water level of 1.6 m and a constant flow rate (scheduled demand) of 5 m^3/s by the orifice offtake. The disturbance was not introduced initially. The operation period for the gate is 5 minutes and the predictive horizon length is 8 hours, so there are 96 operation periods ($K_F = 96$).

The disturbance

In order to evaluate the algorithm performance, we introduce a disturbance in the canal, which is unknown for the water manager. We will consequently get variations between the measured/simulated water level and the desired water level once the disturbance is introduced into the canal. In that way, the disturbance is introduced from 30 minutes after the starting time to the test ending, with a flow rate of 2 m^3/s.

Table 1 | Canal features

Pool number	Pool length (km)	Bottom slope (%)	Side slopes (H:V)	Manning's coefficient (n)	Bottom width (m)	Canal depth (m)
1	2.5	0.1	1.5:1	0.025	5	2.5

Table 2 | Sluice gate, pump station, and orifice offtake features (canal structures)

Number of control structure or checkpoint	Gate discharge coefficient	Gate width (m)	Gate height (m)	Step (m)	Discharge coef./ diameter orifice offtake (m)	Orifice offtake height (m)	Lateral spillway height (m)	Lateral spillway width (m)/ discharge coefficient
0	0.61	5.0	2.0	0.0	–	–	–	–
1	–	–	–	–	2/0.85	0.8	2.0	5/1.99

Table 3 | Overall and functional constraint values in single canal pool

	U_{min} (%)	U_{max} (%)	dU_{max}/d_0U_{max} (%)
One pool canal	0.05	90	2.5

Constraints

The constraints of the problem were imposed on gate opening and gate movement (control action variable). These constraints were exactly the same in both control algorithms. In that way, the gate opening is not greater or smaller than U_{max} or U_{min}, respectively, and the gate movements between successive operation periods (dU_{max}) are physically acceptable as well as the initial gate movement for different prediction horizons ($d_0U_{max} = dU_{max}$), see Bonet (2015) (Table 3).

Results

We represent with a discontinuous line the water level at the checkpoint obtained with the overall control diagram with GoRoSoBo simplified. On the other hand, the results obtained with the overall control diagram using GoRoSoBo are shown with a continuous line in Figure 6. We can also compare both gate trajectories in Figure 7.

The flow condition in the canal is steady state at the beginning of the test, because the scheduled demand is a constant flow rate for the irrigation cycle and the sluice gate position remains fixed. After the first 30 minutes (1,800 s), a disturbance is introduced into the system, and the water level decreases 25 cm at the checkpoint at 2,700 s (Figure 6).

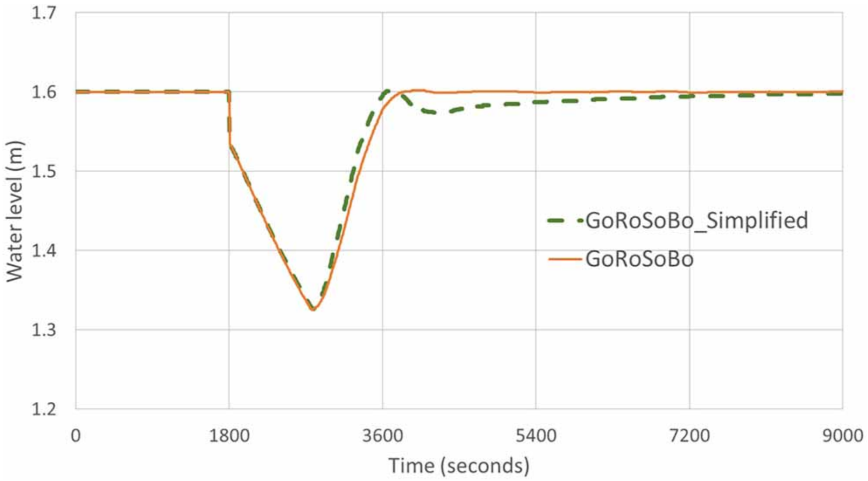

Figure 6 | Water level at the checkpoint by GoRoSoBo (continuous line)/GoRoSoBo simplified (discontinuous line).

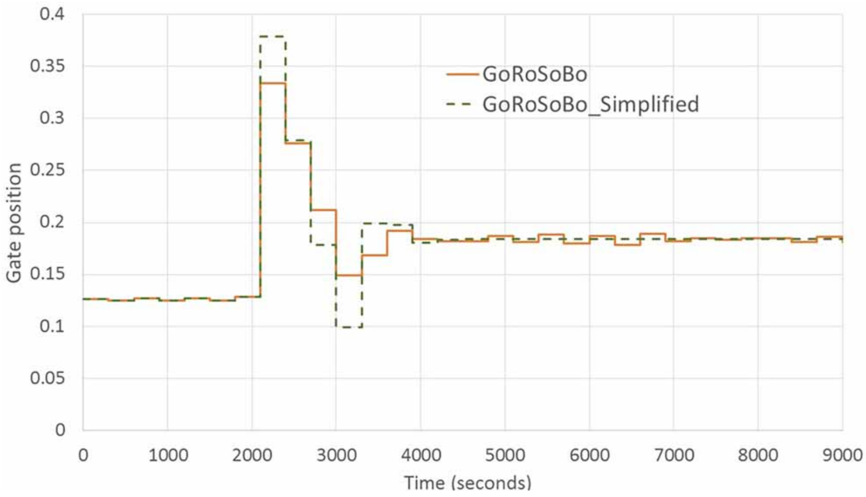

Figure 7 | Gate trajectory obtained by GoRoSoBo (continuous line)/GoRoSoBo simplified (discontinuous line).

On the other hand, the water level is not going to increase at the checkpoint up to three operation periods later (2,700 s) in both algorithms, due to the wave delay from gate to checkpoint. Once the wave arrives at the checkpoint, the water level increases quickly reaching the target level of 1.6 m at 3,600 s or 3,700 s, regarding GoRoSoBo simplified or GoRoSoBo. The water level reaches the desired value at 3,700 s with GoRoSoBo, but the water level does not reach the desired water level up to 7,200 s with GoRoSoBo simplified.

We can conclude that the gate trajectories obtained with GoRoSoBo are more accurate than GoRoSoBo simplified (Figure 7) and we reach the water level target before using GoRoSoBo than GoRoSoBo simplified; however, the GoRoSoBo simplified results are quite accurate considering Figure 7.

Regarding computational time, GoRoSoBo simplified reduces almost five times the initial computational time of GoRoSoBo. Whereas GoRoSoBo needs 20 seconds to calculate the gate trajectories for a prediction horizon of 8 hours using three iterations (every operation period), GoRoSoBo simplified only needs 4 seconds for the same prediction horizon (every operation period). The computational time reduction is dependent on the number of reaches, gates, cross sections, weirs, and pumps so a computational time reduction of almost five times regarding a simple geometry canal is a useful control algorithm upgrade. The PC used in all tests is a HP i5 4 Gb Ram and 128 Gb SSD.

Numerical example: a canal with multiple disturbances at the same time: ASCE test cases

In this numerical example, we introduce the test cases which were proposed by the ASCE committee to evaluate control algorithms. Two canals are evaluated by the ASCE Task Committee (Clemmens *et al.* 1998) for several scenarios. Each canal has eight pools separated by undershot sluice-gates. All eight canal pools have been discretized and numbered in the direction of flow from upstream to downstream and they have eight checkpoints and eight gates. In both canals, there are gravity outlet orifices at the downstream end of each pool, only in the case of canal 2 there is a pump station at the end of the last pool. The ASCE committee proposes four cases to test feedback controllers in real time, two cases for the Corning canal and two cases for the Maricopa Stanfield. We only test one case in each canal, the most difficult cases (with important unscheduled flow changes). Furthermore, we have analyzed the results obtained with GoRoSoBo and GoRoSoBo simplified showing the computation time required for each one. The canal geometry, scenarios' flow conditions, and time sampling have not been published in this paper in order

Table 4 | Overall and functional constraints values in test cases

	U_{min} (%)	U_{max} (%)	dU_{max}/d_0U_{max} (%)
Maricopa Stanfield	2	90	5
Corning canal	0.5	90	5

not to introduce too much information. All this descriptive test information can be seen in Clemmens *et al.* (1998).

Constraints

The constraints of the problem were imposed to gate position/opening and gate movements (control action variable) and these constraints were the same in both control algorithms. The constraints are included as a percentage of the gate height imposed on the gate positions (Table 4).

We show the results (water level at checkpoints and gate trajectories) obtained in each test in the next figures (Figures 8–11). With the aim of facilitating understanding of the results, we only show the water level at two checkpoints in every figure as well as the gate trajectory at checkpoints. In order to reduce the figure captions, we call in advance GoRoSoBo as GRSB and GoRoSoBo simplified as GRSB_S.

Test case 1-2 (Maricopa Stanfield)

In such a case, all water delivery changes (unscheduled deliveries) are relevant according to Table 5.

Considering the flow change at checkpoint 8, the unscheduled deliveries are more significant at checkpoint 8 where the flow rate changes from 0.9 m^3/s to 0.6 m^3/s (33%). In this case, gate 8 must close in order not to cause overflow in the last pool and other gates also close decreasing the flow to reduce the water level at other checkpoints (see Figure 9).

It is important to remark that the unscheduled water deliveries are relevant in all targets but especially at target 8, because in just one operation period, the water level at checkpoint 8 increases from 0.8 m to 1.05/1.08 m using

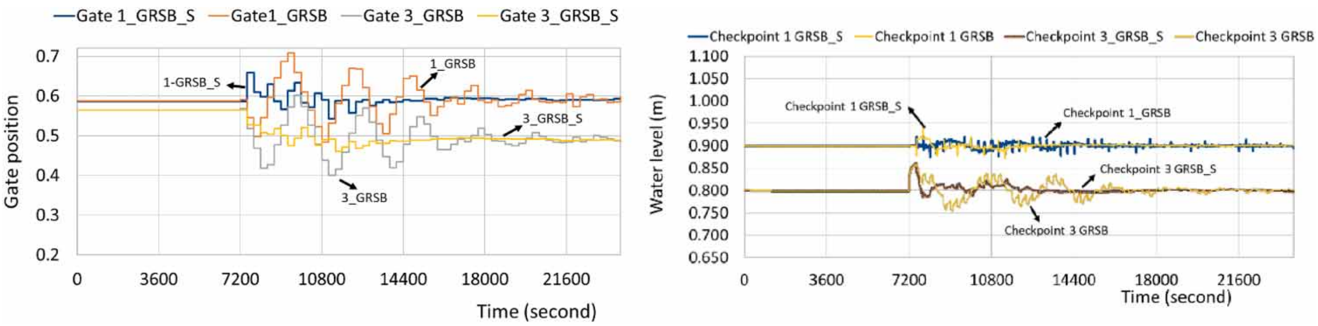

Figure 8 | Gate trajectories 1 and 3 (left), water level at checkpoints 1 and 3 (right) in Test case 1-2 with GRSB and GRSB simplified.

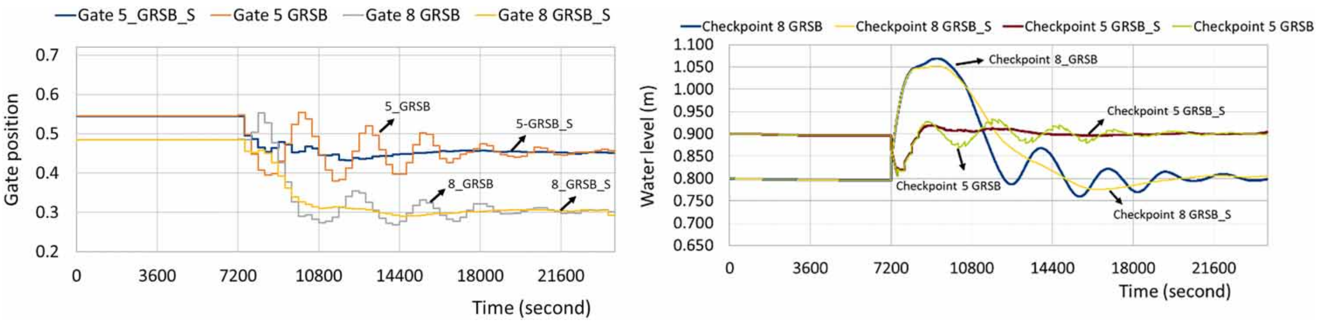

Figure 9 | Gate trajectories 5 and 8 (left), water level at checkpoints 5 and 8 (right) in Test case 1-2 with GRSB and GRSB simplified.

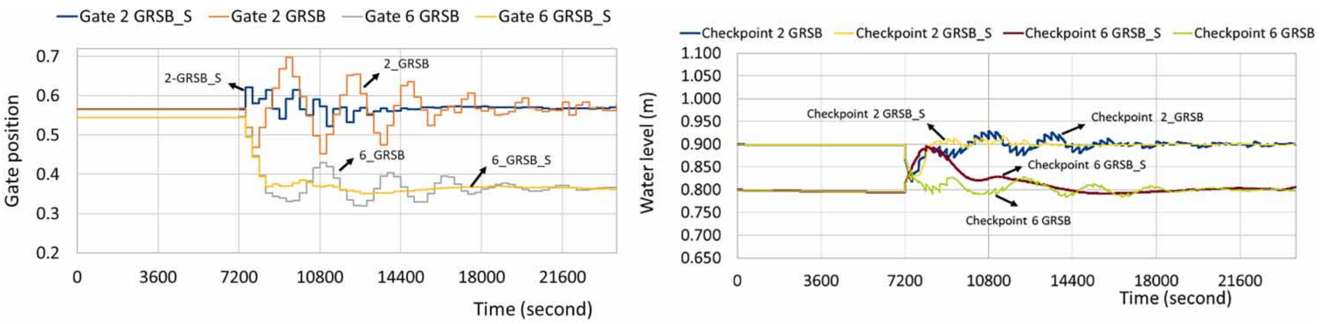

Figure 10 | Gate trajectories 2 and 6 (left), water level at checkpoints 2 and 6 (right) in Test case 1-2 with GRSB and GRSB simplified.

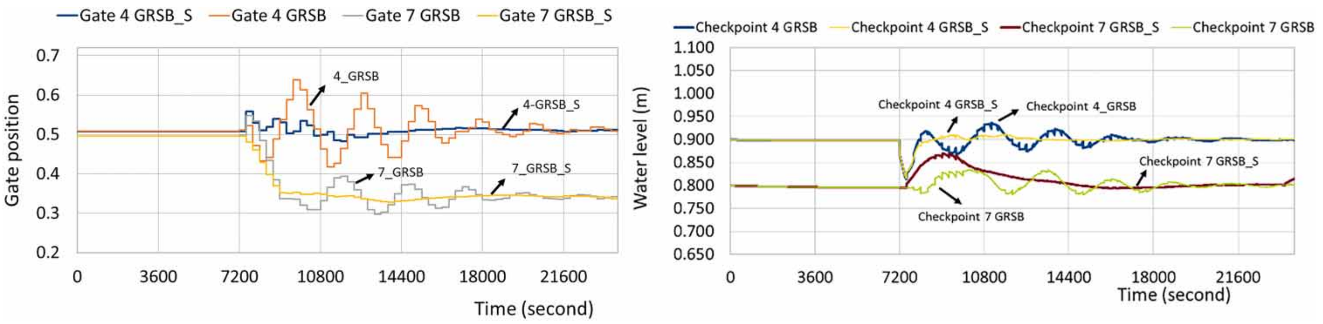

Figure 11 | Gate trajectories 4 and 7 (left), water level at checkpoints 4 and 7 (right) in Test case 1-2 with GRSB and GRSB simplified.

GoRoSoBo simplified or GoRoSoBo, respectively (Figure 9). In a general view, we observe that water level reaches the target level at all checkpoints in GoRoSoBo before GoRoSoBo simplified. The water level oscillation at checkpoints is longer in GoRoSoBo than GoRoSoBo simplified, because the gate movements are smooth in GoRoSoBo simplified as this algorithm does not update the HIM matrix (see Figures 8–11).

Table 5 | Initial and unscheduled offtake changes in Test case 1-2

Pool number	Offtake initial flow (m^3/s)	Check initial flow (m^3/s)	Unscheduled offtake changes at 2 hours (m^3/s)	Check final flow (m^3/s)
Heading	–	2.0	–	2.0
1	0.2	1.8	–	1.8
2	0.0	1.8	0.2	1.6
3	0.4	1.4	−0.2	1.4
4	0.0	1.4	0.2	1.2
5	0.0	1.4	0.2	1.0
6	0.3	1.1	−0.1	0.8
7	0.2	0.9	–	0.6
8	0.9	0.0	−0.3	0.0

The reduction in computational time using GoRoSoBo simplified in respect to GoRoSoBo was almost 16 times. Whereas GoRoSoBo needs 720 seconds to calculate the gate trajectories for a prediction horizon of 2 hours and 30 minutes using three iterations (for every operation period), GoRoSoBo simplified only needs 45 seconds for the same prediction horizon (for every operation period).

Test case 2-2 (Corning canal)

Test case 2-2 is the most difficult test due to the significant unscheduled water deliveries in all targets, according to Table 6.

GoRoSoBo calculates in a more accurate way the gate trajectories (see Figures 12 and 13) than GoRoSoBo, because GRSB reaches the target level before GRSB simplified (see Figures 14 and 15). GoRoSoBo uses the HIM updated at each time, but GRSB simplified only uses a HIM matrix at an initial steady state, thus the accuracy of

Table 6 | Initial and unscheduled offtake changes in Test case 2-2

Pool number	Offtake initial flow (m^3/s)	Check initial flow (m^3/s)	Unscheduled offtake changes at 2 hours (m^3/s)	Resulting check flow (m^3/s)
Heading	–	13.7	–	2.7
1	1.7	12.0	−1.5	2.5
2	1.8	10.2	−1.5	2.2
3	2.7	7.5	−2.5	2.0
4	0.3	7.2	–	1.7
5	0.2	7.0	–	1.5
6	0.8	6.2	−0.5	1.2
7	1.2	5.0	−1.0	1.0
8	0.3 + 2.0*	2.7	−2.0*	0.7

GRSB is understandable. The maximum water level at all checkpoints is higher in GoRoSoBo simplified than GoRoSoBo, even more at checkpoints 1, 3, 4, and 5, in which the water level is almost 10 cm higher (see Figures 14 and 15).

The gate movements are almost zero during the last 2 hours of the test, because the water levels at checkpoints are close to the desired values, with a maximum error around 5 mm in both algorithms.

The reduction in computational time using GoRoSoBo simplified with respect to GoRoSoBo was almost eight times. Whereas GoRoSoBo needs 390 seconds to calculate the gate trajectories for a prediction horizon of 2 hours and 30 minutes using three iterations (for every operation period), GoRoSoBo simplified only needs 52 seconds for the same prediction horizon (for every operation period).

GRSB introduces important movements at the canal gates in comparison with GRSB simplified (Figures 12 and 13). GRSB algorithm uses an accurate HIM matrix version at every operation period (15 minutes for Corning canal and 5 minutes for Maricopa Stanfield, both operation periods proposed by the ASCE committee), instead, GRSB simplified only uses a steady state HIM matrix version (obtained before the irrigation cycle started). In that sense, GRSB can adapt the gate trajectories according to

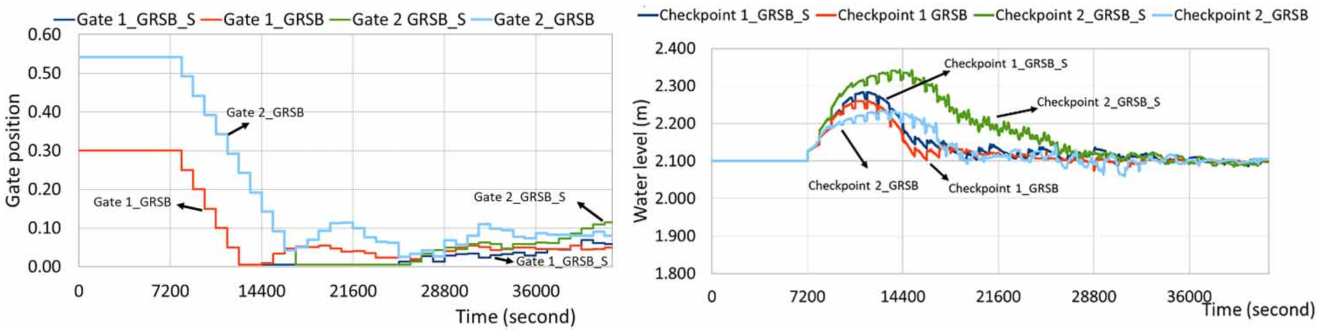

Figure 12 | Gate trajectories 1 and 2 (left), water level at checkpoints 1 and 2 (right) in Test case 2-2 with GRSB and GRSB simplified.

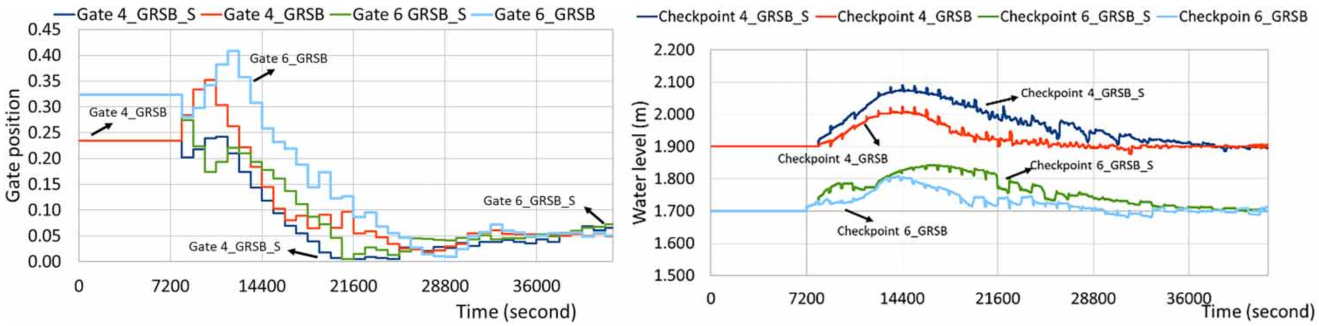

Figure 13 | Gate trajectories 4 and 6 (left), water level at checkpoints 4 and 6 (right) in Test case 2-2 with GRSB and GRSB simplified.

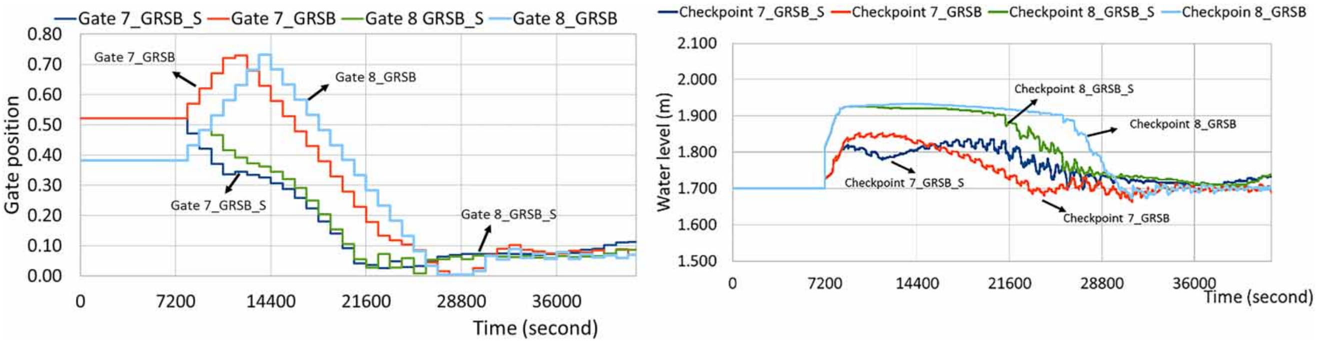

Figure 14 | Gate trajectories 7 and 8 (left), water level at checkpoints 7 and 8 (right) in Test case 2-2 with GRSB and GRSB simplified.

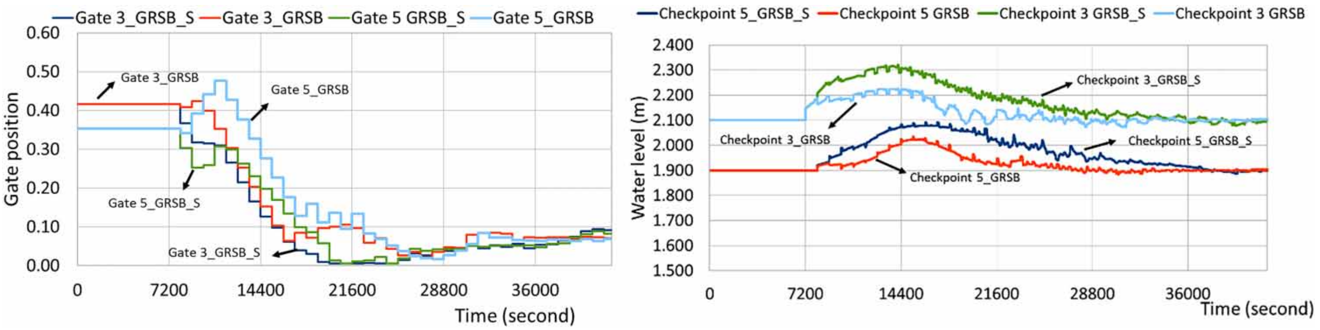

Figure 15 | Gate trajectories 3 and 5 (left), water level at checkpoints 3 and 5 (right) in Test case 2-2 with GRSB and GRSB simplified.

water level error faster, because its predictive model (HIM matrix) is more accurate than GRSB simplified and HIM matrix is always updated. Instead, GRSB simplified needs to apply a large Marquardt coefficient in order to converge upon a feasible solution calculating smooth gate trajectories less accurate than the GRSB solution.

Performance indicators

Particular indicators were introduced by ASCE Task Committee to compare different control algorithms (Clemmens *et al.* 1998), in order to evaluate them and judge the controller's ability to deliver water. This is the reason why the ASCE Task Committee devised these test cases.

We compare the performance indicators obtained with GoRoSoBo and GoRoSoBo simplified in test cases with the performance indicators obtained with other control algorithms such as CLIS (Liu *et al.* 1998), PILOTE (Malaterre 1995), and ATV-PID control (Ocampo-Martinez & Negenborn 2015) (see Tables 7 and 8). The CLIS is based on an inverse solution method of the Saint-Venant equations, and it is designed for the automation of demand-oriented systems. The PILOTE is a LQR closed-loop controller and it is obtained from the steady-state solution of the Riccati equation, a Kalman filter is used to reconstruct the state variables and the unknown perturbations from a reduced number of observed variables. The ATV-PID control combines a PID controller tuned by the ATV method, which is a classical automatic tuning method to identify important characteristics of the response of a monovariable process using a simple relay experiment. The ATV-PID algorithm performs regarding several operational options depending on the type of control action variables:

1. The flow with check gates acting as pumps, hereafter noted option *P*.
2. The flow with discharge inversion for calculating the opening of the check gates at each control time step, hereafter noted option *Q*.
3. The gates opening W, hereafter noted option *W*.

Table 7 | The performance indicators obtained in Test case 1-2 (Maricopa Stanfield)

	Test case 1-2 tuned-unscheduled							
	MAE **(%)**		***IAE*** **(%)**		***StE*** **(%)**		***IAQ*** **(m^3/s)**	
	12–24 h		**12–24 h**		**12–24 h**		**12–24 h**	
	Max	**Ave.**	**Max**	**Ave.**	**Max**	**Ave.**	**Max**	**Ave.**
CLIS	34.5	14.2	5.0	2.0	3.6	1.1	0.2	0.1
PILOTE	43.0	24.9	9.2	5.2	11.2	2.9	2.9	1.4
GoRoSoBo	33.5	10.3	5.0	1.2	2.1	0.7	6.7	3.6
GoRoSoBo simplified	32.9	11.6	5.3	1.7	2.3	0.9	3.9	2.0

Table 8 | The performance indicators obtained in Test case 2-2 (Corning canal)

	Test case 2–2 tuned-unscheduled							
	MAE **(%)**		***IAE*** **(%)**		***StE*** **(%)**		***IAQ*** **(m^3/s)**	
	12–24 h		**12–24 h**		**12–24 h**		**12–24 h**	
	Max	**Ave**	**Max**	**Ave.**	**Max**	**Ave.**	**Max**	**Ave.**
CLIS	21.1	14.9	7.6	2.8	0.7	0.4	9.7	5.5
PILOTE	34.2	17.1	10.6	7.1	8.8	4.3	10.4	6.1
ATV-PID (Q: $\alpha = 1$: Up to Dn)	40	–	17.5	–	19	7	–	–
ATV-PID (P: $\alpha = 1$: Up to Dn)	–	15.2	–	7.6	–	–	–	–
ATV-PID (P: $\alpha = 0$: Up to Dn)	–	–	–	–	–	–	2.1	0.35
GoRoSoBo	13.6	7.8	6.3	2.1	0.8	0.5	15.2	11.7
GoRoSoBo simplified	13.3	10.1	5.6	3.4	3.2	1.3	10.1	7.1

Regarding ATV-PID control, only options with best performance indicator results (Ocampo-Martinez & Negenborn 2015) have been included in Table 8.

The *MAE* is the maximum deviation of the controlled water level at the checkpoint with regard to the desired water level. From the values obtained by GoRoSoBo and GoRoSoBo simplified with this performance indicator, we can conclude that GoRoSoBo simplified shows the best results in *MAE* maximum in all tests and GoRoSoBo shows the best results in *MAE* average in all tests.

The *IAE* is the integrated deviation of controlled water level on the target water level. GoRoSoBo shows the best values of maximum *IAE* in one case and GoRoSoBo simplified in the other case. GoRoSoBo shows the best values of average *IAE* in two cases, whereas GoRoSoBo simplified is second in one case.

The *StE* is the deviation of controlled water level at steady state over target water depth, so we only consider the last 2 hours of the irrigation cycle to calculate this performance indicator. GoRoSoBo obtained the best results in Test case 1-2, whereas GoRoSoBo simplified was very close to these results. Instead, CLIS obtained the best results in Test case 2-2.

The *IAQ* is an indicator relative to the wear and tear of the check gates and the best configurations in *IAQ* are underreacting controllers (Ocampo-Martinez & Negenborn 2015). GoRoSoBo and GoRoSoBo simplified show the worst values on this performance indicator, with the ATV-PID showing the best results. When a control algorithm has a single objective, which is to ensure the water level reaches the desired water level, the gate trajectories are only calculated for that objective so there are performance indicators that benefit more (such as *MAE*/*IAE*) than others (such as *IAQ*).

Table 9 | Advantages and disadvantages between GRSB and GRSB simplified

GRSB		**GRSB simplified**	
Advantage	**Disadvantage**	**Advantage**	**Disadvantage**
Performance indicators show the best results	Long computation time between sampling intervals to build the HIM matrix	In this case, HIM matrix is not calculated between intervals' sampling, so GRSB simplified could be used in case of short sampling intervals to build the HIM matrix	GRSB simplified does not show the best values on performance indicators
GRSB calculates very accurate gate trajectories reaching quickly the target water level	GRSB is the best option for specific cases (long sampling intervals and not complex canal geometry)	Performance indicators are good enough in comparison with other control algorithms	After several irrigation cycles, HIM matrix should be updated to keep up the dynamics of the system

In that sense, when significant flow changes are introduced into the system, it is necessary for quick changes in gate trajectories to recover the desired water level at checkpoints as soon as possible. For this reason, increasing the constraint value in gate movements to obtain better values of the *IAQ* indicator and worsen the *MAE* and *IAE* indicators is not our principal objective.

CONCLUSIONS

The GoRoSoBo and GoRoSoBo simplified algorithms are able to find the optimum gate trajectory during a predictive horizon from demand deliveries, initial gate trajectories, desired water level vector, disturbances, and the current canal state (obtained from CSE algorithm). These input data are introduced into GoRoSoBo or GoRoSoBo simplified in order to recalculate the optimum gate trajectory to keep the water level close to the desired water level at checkpoints.

Both algorithms use the Lagrange-Newton method (SQP method) to solve a constrained optimization problem. This method is considered the most efficient when one has compiled the Jacobian matrix and the Hessian matrix which are used in the computation of the gate trajectories.

The introduction of constraints is absolutely necessary to ensure stability in our optimized problem, due to inherent instability in the unconstrained problem. Furthermore, the Hessian and HIM matrixes are ill-conditioned matrixes, so Marquardt coefficient must be introduced in the diagonal of the matrix improving its condition number and solving convergence problems associated with the method by those matrixes and the water level vector error. In that sense, GoRoSoBo simplified has to increase the Marquardt number (Bonet 2015) in comparison with GRSB in order to avoid convergence problems during the calculating process and reaching a solution.

GoRoSoBo and GoRoSoBo simplified obtained accuracy values in *MAE* and *IAE* performance indexes.

The *IAQ* index is higher in GoRoSoBo than GoRoSoBo simplified and much higher than other controllers proposed. These index results are strongly linked to the main priority of the canal. In the case that the test priority is to keep the water level close to the desired water level at the checkpoints for any disturbance, the gate movements are significant and the flow rate variations through the sluice gate too, so the *IAQ* index will also be significant.

The GoRoSoBo simplified algorithm obtained good results regarding the *STE* index, although the algorithm has great scope for improvement. We could restrict the gate movements when water level error is lower than a certain threshold (we could introduce a dead band).

One of the main problems of our overall control diagram is the computation time. In cases where the predictive horizon is large, the operation period is short and the canal length and the number of checkpoints are significant, the calculation time of the algorithm is too long for operating in real time regarding HIM matrix calculation. A solution to this problem is our new overall control diagram simplified. The computation time is reduced between five and 16 times depending on the test, geometry, time steps, iterations, constraints using the GoRoSoBo simplified instead of GoRoSoBo. The reduction of computation time makes GoRoSoBo simplified compatible for huge canals with several checkpoints and gates, which is common in most real canals. The GoRoSoBo simplified has huge scope for improvement as the algorithm code could be parallelized with CUDA or OpenMP to reduce the CPU time. The best result obtained in all these tests is the accuracy of GoRoSoBo simplified compared with other feedback algorithms such as GoRoSoBo, CLIS, or PILOTE.

A comparative table (Table 9) between GRSB and GRSB simplified is added to evaluate strong and weak points of the algorithms.

SUPPLEMENTARY DATA

The Supplementary Data for this paper is available online at http://dx.doi.org/10.2166/hydro.2019.159.

REFERENCES

Aguilar, J. V., Langarita, P., Rodellar, J., Linares, L. & Horváth, K. 2016 Predictive control of irrigation canals – robust design and real-time implementation. *Water Resources Management* **11**, 3829–3843.

Álvarez, A., Ridao, M. A., Ramirez, D. R. & Sánchez, L. 2013 Constrained predictive control of an irrigation canal. *Journal of Irrigation and Drainage Engineering* **139** (10), 841–854. https://doi.org/10.1061/(ASCE)IR.1943-4774.0000619.

Bautista, E., Clemmens, A. J. & Strelkoff, T. 1997 Comparison of numerical procedures for gate stroking. *Journal of Irrigation and Drainage Engineering* **113** (2), 129–136.

Bonet, E. 2015 *Experimental Design and Verification of A Centralize Controller for Irrigation Canals*. PhD Thesis, UPC, Technical University of Catalonia, Barcelona, Spain.

Bonet, E., Gómez, M., Soler, J. & Yubero, M. T. 2016 CSE algorithm: 'canal survey estimation' to evaluate the flow rate extractions and hydraulic state in irrigation canals. *Journal of Hydroinformatics* **19**, 62–80. doi:10.2166/hydro.2016.014.

Bonet, E., Gómez, M., Yubero, M. T. & Fernández-Francos, J. 2017 GOROSOBO: an overall control diagram to improve the efficiency of water transport systems in real time. *Journal of Hydroinformatics* **19**, 364–384. doi:10.2166/hydro.2017.225.

Chevereau, G. 1991 *Contribution à l'étude de la régulation dans les systèmes hydrauliques à surface libre (Contribution to the Study of the Regulation in Free-Surface Hydraulic Systems)*. Thèse de Doctorat de l'Institut National Polytechnic de Grenoble, France.

Clemmens, A. J. & Wahlin, B. T. 2004 Simple optimal downstream feedback canal controllers: ASCE test case results. *Journal of Irrigation and Drainage Engineering* **130**, 35–46.

Clemmens, A. J., Kacerek, T. F., Grawitz, B. & Schuurmans, W. 1998 Test cases for canals control algorithms. *Journal of Irrigation and Drainage Engineering* **124** (1), 23–30.

Fletcher, R. 1987 *Practical Methods of Optimization*, 2nd edn. John Wiley & Sons, Chichester, UK.

Gill, P. E., Murray, W. & Wright, M. H. 1981 *Practical Optimization*. Academic Press, London, UK.

Gómez, M. 1988 *The Contribution of Case Study of Transient Flow in Sewer Systems*. PhD Thesis, UPC, Barcelona, Spain.

Liu, F., Feyen, J. & Berlamont, J. 1992 Computation method for regulating unsteady flow in open channels. *Journal of Irrigation and Drainage Engineering* **118** (10), 674–689.

Liu, F., Feyen, J., Malaterre, P. O., Baume, J. P. & Kosuth, P. 1998 Development and evaluation of canal automation algorithm CLIS. *Journal of Irrigation and Drainage Engineering* **124** (1), 40–46.

Malaterre, P. O. 1994 *Modelisation, Analysis and LQR Optimal Control of an Irrigation Canal*. PhD Thesis, LAAS-CNRS-ENGREF-Cemagref, Etude EEE n14, France.

Malaterre, P. O. 1995 PILOTE: optimal control of irrigation canals. In: *First International Conference on Water Resources Engineering, Irrigation and Drainage*, August 14–18, 1995, San Antonio, TX, USA.

Marzouki, T. Z. 1989 Le système hydraulique de la rivière larcis (barrages, rivière, périmètres irrigués). Modélisation hydraulique, analyse et amélioration de la gestion (The hydraulic system of the Larcis river (dams, river, irrigated perimeters). Hydraulic modeling, analysis and improvement of management) CEMAGREF, ENGREF, CARA, p 63.

Ocampo-Martinez, C. & Negenborn, R. R. (eds) 2015 *Transport of Water Versus Transport Over Water*. Springer, Switzerland. https://doi.org/10.1007/978-3-319-16133-4.

Soler, J., Gómez, M. & Rodellar, J. 2008 A control tool for irrigation canals with scheduled demands. *Journal of Hydraulic Research* **46**, 152–167.

Wahlin, B. T. & Clemmens, A. J. 2006 Automatic downstream water-level feedback control of branching canal networks: simulation results. *Journal of Irrigation and Drainage Engineering* **132**, 208–219.

Wylie, E. B. 1969 Control of transient free-surface flow. *Journal of the Hydraulics Division* **95**, 347–362.

First received 3 December 2017; accepted in revised form 13 August 2019. Available online 4 October 2019