Multivariable and Distributed LQG Control of a Water Delivery Canal

João M. Lemos¹; Luís F. Pinto²; Luís M. Rato³; and Manuel Rijo⁴

Abstract: This paper addresses the problem of the development of a distributed linear quadratic Gaussian (LQG) controller for a water delivery canal. The control structure proposed relies on a set of LQG control agents interconnected through a communication network. Each of these local control agents controls a canal reach made of a pool and the corresponding downstream gate and receives information (output signal and control moves) only from the corresponding canal reach and the ones that are adjacent to it. An algorithm is proposed to achieve coordinated action of the different local control agents. This distributed control structure is compared with centralized multivariable LQG control. Several aspects with incidence on performance are addressed, including the modification of the quadratic cost to ensure a constraint on closed-loop poles, the use of a nonlinear filter to limit noise effects, and the impact of a quantization commonly forced in gate position. Experimental results obtained in a pilot canal are presented. DOI: 10.1061/(ASCE)IR.1943-4774.0000621. © 2013 American Society of Civil Engineers.

CE Database subject headings: Canals; Water distribution systems.

Author keywords: Water delivery canal; Canal control; Distributed control; Multivariable control; Linear quadratic Gaussian (LQG).

Introduction

Problem Formulation and Motivation

Water delivery canal systems are often spread over wide geographical areas, with actuators and local controllers in isolated spots (Marcola et al. 2005; Cantoni et al. 2007). Furthermore, due to their physical characteristics, the use of pure decentralized control, i.e., structures in which local controllers make decisions about manipulated variables on the basis of pure local measurements without any exchange of information among them, may lead to poor performance or even instability. On the other way, completely centralized control architectures may not only be unfeasible due to the complexity of the transmission network involved, but highly unreliable as well because communication links may be interrupted by hazardous causes.

These features provide a strong motivation for the employment of distributed control in which a network of local control agents act in a coordinated way by negotiating with their neighbors. If designed adequately, distributed control has a number of advantages:

- Simplicity of the design of local controllers, conjugated with good overall performance;
- Increased reliability with respect to failures of either local control agents or communication links; and
- Adequate management of local objectives, which may vary from branch to branch for canals that cross different administrative districts.

The objectives associated with the management and control of water delivery canal systems are manifold and depend on the type of canal and the way it is operated. Usually, saving water is a major aim, with automatic control known to increase canal efficiency (Litrico and Fromion 2009). In some cases, the water is driven to the canal only during the periods of usage, while in other situations of interest a minimum water level must always be ensured, for instance, for environmental reasons. In any case, in irrigation canals the key point is to ensure that the desired amount of water during a specified period of time is available to the farmers that require it, while simultaneously water level constraints are respected, e.g., to avoid overflows that represent water spillage and may damage certain types of canals or to ensure a minimum water level along the canal, in particular near the turnouts. If translated in terms of control system specifications, these requirements mean that the water levels must follow given references with specified transients, and that disturbances due to water extraction by users are rejected.

In the situation considered in this paper, the main concern is to keep the downstream boundary water level of each pool close to specified reference values, rejecting disturbances induced by turn-out consumptions. Hence, local upstream control is used in order to speed up disturbance rejection. Although other schemes could be considered that lead to a more efficient use of water (Litrico and Fromion 2009), the concern in this paper is to demonstrate how a linear quadratic Gaussian (LQG) controller can be modified to act in a distributed setting to regulate water level in a canal with multiple pools.

Literature Review

Centralized LQG control of hydro systems has long been considered (Mays 1997; Weyer 2003; Litrico and Georges 2001;
Bautista et al. 2006). Model predictive control (MPC) provides an approximation to LQG that has the advantage of allowing the incorporation of constraints in an easy way, but requires a higher computational load. Applications of MPC to water delivery canals include both centralized (Begovich et al. 2007) and distributed examples (Igreja et al. 2012). In El Fawal et al. (1998), a decentralized version of LQG for a multicanal system has been obtained in a way that has a tight connection with predictive control. Therefore, the overall system is decomposed in a number of interacting subsystems. A local control agent is then associated with each subsystem in which the manipulated variables are computed by minimizing a receding horizon quadratic cost using the LQG algorithm. The coordination between the controllers of the local subsystems is achieved using the decomposition–coordination approach based on dual optimization (Boyd and Vandenberghe 2004). Accordingly, the manipulated variables are computed at the beginning of each sampling interval by iterating through two steps. First, the decomposition step is performed, which consists for each subsystem of minimizing a Lagrangian formed by augmenting the local quadratic cost with a term embedding a constraint of continuity between different canal reaches, multiplied by Lagrange multipliers. Then, the Lagrange multipliers are updated in the coordination step.

A different approach to obtain distributed versions of LQG consists of imposing a structure to the controller that matches the desired distributed architecture and to compute the optimal gains using a gradient algorithm. In Lewis and Symons (1995, pp. 411–413), the distributed linear quadratic (LQ) regulator problem for a pair of coupled systems is solved by applying an algorithm for optimal control with an a priori imposed structure described in Lewis and Symons (1995, p. 370). Estimates of the gradient based on the adjoint equation are suggested in Murtensson and Rantzer (2010) for deterministic plants and in Rantzer and Murtensson (2009) for stochastic plants.

Another possibility to coordinate distributed controllers is to apply game theory concepts (Maestre et al. 2011). In this case, each control agent plays a game with its neighbors in which each player tries to optimize a local cost by assuming knowledge of the others’ control moves. In Zhang and Li (2007), by iterating the negotiation process, a Nash equilibrium is attained (Webb 2006).

This last approach is used in this paper to develop a distributed version of LQ control. The controller agents play a game among themselves by trying in successive iterations to minimize their respective quadratic costs, while using for feedback action the value of the manipulated variables of their neighbors that is computed in the previous round. Although convergence to a stabilizing controller is not in general ensured by the resulting Nash equilibrium, in the class of systems considered in this paper, stability is attained together with a performance that is close to the optimum. Furthermore, this approach has the advantage of requiring a much lower computational load than algorithms that rely on dual optimization and is the one followed in this paper.

**Canal Description**

The work reported in this paper was performed at the experimental canal of Núcleo de Hidráulica e Controle de Canais (Universidade de Évora, Portugal), for which several studies previously conducted are available (Litrício et al. 2005; Wertz et al. 2005; Lemos et al. 2009).

As shown in Fig. 1, the canal has four pools with a length of 35 m each, separated by three undershoot gates, with the last pool ended by an overshot gate. The maximum nominal design flow is 0.09 m³ s⁻¹. Each pool has an outfall near the downstream boundary that allows drawing of the water flows that simulate water consumption by farmers.

Water level sensors are installed downstream of each pool, being placed inside a stilling well connected to the bottom of the canal by a pipe. The water level sensors measure values between 0 and 900 m, a value that corresponds to the canal bank. The nomenclature is as follows (Fig. 1): For pool number i, i = 1, …, 4; the downstream level is denoted yᵢ; and the opening of gate i is denoted uᵢ. Pool number i ends with gate number i. The turnout flow of pool i is denoted Qᵢ.

The canal is supplied by gravity from a reservoir through a small pipe, where an electromagnetic flowmeter and a motorized flow control valve (Monovar type) are installed. The flow control valve is controlled automatically by a proportional-integral controller in order to maintain a predefined inflow Q₀ at the head of the canal. The actuators of each gate are controlled by local programmable logic controllers (PLCs). In turn, local PLCs are connected to a central computer with a supervisory control and data acquisition (SCADA) system. In order to allow fast controller prototyping, it is possible to connect via a wireless network to the central computer other computers that run MATLAB programs. These programs read sensor signals and manipulate gate commands whose values are computed by the control algorithms. Further details on the information network infrastructure can be found in Duarte et al. (2011).

**Linear Models**

The dominant dynamic behavior of water delivery canals can be modeled by the Saint-Venant equations, a pair of nonlinear partial differential equations that embed mass and momentum conservation (Cunge et al. 1980). Although this representation is infinite dimensional, around a given operating point the water level dynamics can be approximated by finite dimensional linear state-space (FDLSS) models. In order to obtain such models, which are needed for linear control design, there are several possibilities. The first is to approximate numerically the Saint-Venant equations, either with a difference scheme such as the Preisman method (Litrício and Fromion 2009) or orthogonal collocation. Another possibility consists in obtaining a transfer function directly by manipulating the Saint-Venant equations (Litrício et al. 2005; Oioa et al. 2005). Although the last procedure has the advantage of providing a pencil of linear models for a wide range of operating conditions defined by an interval of values of water level and flow, it requires the
estimation of physical parameters that may be difficult to obtain and, furthermore, it ignores the dynamics of elements that are very hard to capture from first principles modelling.

To overcome these problems one may resort to system identification performed from plant data, eventually including static nonlinearity compensation to improve the linearity of the system, for instance, using as manipulated variables flows instead of gate positions (Weyer 2001; Euret and Weyer 2007). This method has the drawback of representing the canal dynamics only around a given operating equilibrium point, but the advantage of capturing the tight approximation to the actual plant dynamics in the frequency range that is significant to controller design.

According to the paper objectives, the dynamics of the canal considered are represented in this paper by a finite dimensional linear state-space model, built by concatenating models for each of the pools and assuming that adjacent pools interact only through their manipulated inputs. Individual pool models are identified from plant data obtained by perturbing an equilibrium state with a pseudorandom binary signal (PRBS). The data collected in these conditions are then filtered by a Butterworth low pass filter and used to compute least-squares estimates of the parameters of an autoregressive with exogenous input (ARX) model per pool. A state space of order 4 is finally obtained for each pool by converting the ARX models to this representation.

Therefore, in a distributed control framework the canal is assumed to be decomposed in N local subsystems denoted S_i, i = 1, \ldots, N, that are coupled only through their inputs and form a serial chain, i.e., system i interacts only with systems i-1 and i+1 whenever they exist (there are no systems 0 or N+1). In this framework, each subsystem S_i is modeled by the state space representation

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \Gamma_i \delta_i(k)$$  \hspace{1cm} (1)

with the output equation

$$y_i(k) = C_i x_i(k)$$ \hspace{1cm} (2)

where k \geq 0 is a nonnegative integer that denotes discrete time; x_i \in \mathbb{R}^n is the state of subsystem S_i; u_i \in \mathbb{R} is the manipulated variable; y_i \in \mathbb{R} is the measured output; \delta_i \in \mathbb{R}^3 is a vector-accessible disturbance; and A_i \in \mathbb{R}^{n \times n}, B_i \in \mathbb{R}^{n \times 1}, C_i \in \mathbb{R}^{1 \times n}, and \Gamma_i \in \mathbb{R}^{n \times 3} are matrices that define the parameterization of the model of subsystem S_i. It is assumed that (A_i, B_i) is reachable and that (C_i, A_i) is detectable, assumptions that are checked after the model is identified.

It is remarked that the state x_i has no simple physical interpretation in terms of water levels or discharges. Its precise definition is through Eqs. (1) and (2), which relate the state with the variables that have direct physical interpretation u_i (gate position) and y_i (upstream gate water level).

The vector of accessible disturbances is defined by

$$\delta(k) = \begin{bmatrix} u_{i-1}(k) \\ u_{i+1}(k) \\ Q_i(k) \end{bmatrix}$$ \hspace{1cm} (3)

where Q_i = turnout flow that acts as a disturbance on the subsystem S_i.

The model used to design the multivariable controller results from the concatenation of the models defined by Eqs. (1) and (2) obtained for the different pools.

**LQG Control with Pole Constraints**

When using centralized multivariable LQG, a central computer receives all the system outputs (the four downstream pool levels) and computes all the manipulated variables (the four gate positions). Fig. 1 shows the canal interconnection with a centralized multivariable LQG controller. The way this computation is made is explained subsequently. First, the equations for designing local controllers for distributed control are presented. It is then shown how to modify them to design a centralized LQG controller.

**Control Algorithm**

To each subsystem S_i associate a control law that minimizes the quadratic cost:

$$J_i(u) = \frac{1}{2} \sum_{k=1}^{\infty} \left[ x_i^T(k) Q_i x_i(k) + u_i^T(k) R_i u_i(k) \right]$$  \hspace{1cm} (4)

where R_i = \rho_i > 0 is a positive weight in the control action, and

$$Q_i = \begin{bmatrix} C_i^T C_i & 0 \\ 0 & 1 \end{bmatrix}$$ \hspace{1cm} (5)

when a series integrator is included and Q_i = C_i^T C_i when it is not. Using the necessary conditions for optimality in the absence of set constraints given by Goodwin et al. (2005, pp. 65-68), it is concluded that the control law is given by

$$u_{opt,i}(k) = -K_{LQG} x_i(k) + u_{ff,i}(k)$$ \hspace{1cm} (6)

in which the state feedback gain is

$$K_{LQG} = - (\rho_i + B_i^T P_i B_i)^{-1} B_i^T P_i A_i$$ \hspace{1cm} (7)

and P_i satisfies the algebraic Riccati equation

$$P_i = A_i^T P_i \left[ I + \frac{1}{\rho_i} B_i B_i^T P_i \right]^{-1} A_i + Q_i$$ \hspace{1cm} (8)

The feedforward term is

$$u_{ff,i}(k) = (\rho_i + B_i^T P_i B_i)^{-1} B_i^T (g_i - P_i C_i \delta_i)$$ \hspace{1cm} (9)

and the vector g_i satisfies the linear algebraic equation

$$M_i g_i = A_i (\delta_i)$$ \hspace{1cm} (10)

where

$$M_i = I + A_i^T P_i \left[ I + \frac{1}{\rho_i} B_i B_i^T P_i \right]^{-1} B_i B_i^T - A_i^T$$ \hspace{1cm} (11)

and

$$A_i (\delta_i) = -A_i^T P_i \left[ I + \frac{1}{\rho_i} B_i B_i^T P_i \right]^{-1} \Gamma \delta_i$$ \hspace{1cm} (12)

When using centralized multivariable control, the previous formulas may also be used by dropping the index i, using the canal multivariable model that results from concatenating the models for the different pools, and making

$$R = \begin{bmatrix} \rho_1 & 0 & 0 \\ 0 & \rho_2 & 0 \\ 0 & 0 & \rho_3 \end{bmatrix}$$ \hspace{1cm} (13)

In the situation in which the state x_i is not available for direct measure, it is replaced by an estimate generated locally with a Kalman filter, where the assumed process noise variances q are selected according to the loop transfer recovery (LTR) technique (Zhang and Freudenberg 1993). Although it is known that LQ
control yields good stability margins, low values of $q$ may yield significant deviations from its loop gain. This deviation may lead to a degradation of stability margins. By considering the process noise variance $q$ entering the Kalman filter as a tuning parameter and increasing it, the LQG loop transfer gain approaches the LQ loop transfer gain, recovering the LQ stability margins.

In order to constrain the closed-loop poles to lie inside a circle of radius $1/\alpha$, the cost (Eq. (4)) is modified by multiplying each term of the sum by $\alpha^2$. (Franklin et al. 1998). The minimization of this modified cost is performed in a simple way by replacing $A_t$ by $\alpha^2 A_t$ and $B_t$ by $\alpha B_t$.

**Experimental Results**

Fig. 2 shows experimental results obtained with centralized multivariable LQG control. In these plots as well as in all the other experimental plots, the reference is easily distinguishable from the signal because it is made of straight lines. The sampling interval is 2 s and the value of the manipulated variable weights are $\rho_1 = 30 \times 10^3$, $\rho_2 = 20 \times 10^3$, and $\rho_3 = 5 \times 10^3$. These results can be improved by including a nonlinear filter in the level signals feedback to the controller. The filter is such that it only changes its output when the input changes for more than 3 mm. Although the filter limits the tracking precision, it prevents the controller from overactuation.

Fig. 3 shows another set of experimental results obtained with centralized multivariable LQG control, but in which the output nonlinear filter has been included and $\alpha = 1.009$. Comparing these records with the corresponding ones of Fig. 2, it is concluded that the response overshoot has been reduced as well as the gate movement. In both cases, a first-order linear prefilter is applied to the reference to reduce the overshoot in response to a reference step change.

In addition, to show the importance of including the nonlinear output filter and the constraint on the poles, the results of experiments 1 and 2 establish a baseline with which the distributed LQG algorithm proposed is compared.

**Distributed Control**

The structure of distributed control is shown in Fig. 4. To each subsystem $i$ in which the canal is divided, a local controller $C_i$ is associated that computes the position of gate $G_i$. These computation results from a negotiation with neighbor controllers that use the LQG control law previously presented. The coordination algorithm that embeds this negotiation is explained in the next section.

---

**Fig. 2.** Experiment 1: Centralized multivariable LQG control; experimental results with the basic algorithm

**Fig. 3.** Experiment 2: Centralized multivariable LQG control with nonlinear output filter and pole constraint ($\alpha = 1.009$); experimental results
Fig. 4. Canal interconnection with a network of distributed LQG controllers

**Coordination Algorithm**

At the beginning of each sampling interval \( k \), a sequence of virtual control moves \( \bar{u}_{i,j}(k), j = 1, \ldots, j_{\text{max}} \) that result from the negotiation among neighbor control agents is computed. Only the final virtual move \( \bar{u}_{i,j_{\text{max}}}(k) \) is actually applied to the plant. Set

\[
 j = 0
\]

and, for all \( i \),

\[
 \bar{u}_{i,0}(k) = \bar{u}_{\text{opt},i}(k - 1)
\]

and execute in a recursive way the following cycle in the index variable \( j \):

1. Set \( j = j + 1 \);
2. For all control agents \( i \) compute \( \bar{u}_{i,j}(k) \) using Eqs. (6)-(12) in which

\[
 \delta_{i,j} = \begin{bmatrix} \bar{u}_{i-1,j-1}(k) \\ \bar{u}_{i-1,j-1}(k) \\ \vdots \\ \bar{u}_{i,j-1}(k) \\ \bar{u}_{i,j-1}(k) \\ \bar{u}_{i,j-1}(k) \\ \bar{u}_{i,j-1}(k) \end{bmatrix}
\]

3. If \( j \) is equal to a maximum prescribed value \( j_{\text{max}} \), then go to Step 4. Otherwise, go to Step 1.
4. For all \( i \) set

\[
 \bar{u}_{\text{opt},i}(k) = \bar{u}_{i,j_{\text{max}}}(k)
\]

It is remarked that the definition of the disturbance \( \delta_{i,j} \) by Eq. (16) means that two different feedforward terms are considered: Two terms are related with the coordination of the local control move with the control agents of the neighbor pools and one term is related with anticipation of the disturbance effect caused by the turnouts in the pool considered.

Define

\[
 \bar{u}_{i,j}(k) = [\bar{u}_{i,1}(k) \ \bar{u}_{i,2}(k) \ \bar{u}_{i,3}(k) \ \ldots \ \bar{u}_{i,j}(k)]^T
\]

The previous coordination algorithm is equivalent to propagate \( \bar{u}_{i,j}(k) \) using the following linear difference equation in \( j \):

\[
 \bar{u}_{i,j+1}(k) = \Xi \bar{u}_{i,j}(k) + \Upsilon
\]

where

\[
 \Xi = \begin{bmatrix} 0 & \beta_1 & \beta_2 & \ldots & \beta_{\text{max}} \\ \beta_1 & 0 & \beta_2 & \ldots & \beta_{\text{max}} \\ \beta_2 & \beta_3 & 0 & \ldots & \beta_{\text{max}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_{\text{max}} & \beta_{\text{max}} & \beta_{\text{max}} & \ldots & 0 \end{bmatrix}
\]

and

\[
 \Upsilon = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{\text{max}} \end{bmatrix}
\]

with

\[
 \gamma_i = \beta_i \Gamma_i d_i + (\rho_1 + B_i^T P_i B_i)^{-1} B_i^T P_i
\]

As such, the algorithm will converge if the spectral radius of \( \Xi \) satisfies the condition

\[
 \max |\lambda(\Xi)| < 1
\]

where \( \lambda(\Xi) \) denotes the eigenvalues of \( \Xi \). In a game theory framework the resulting equilibrium is a Nash equilibrium (Webb 2006). Although there are plants for which the Nash equilibrium does not ensure stability of the whole system, this is not the case for the application considered in this paper.

**Experimental Results**

The most important tuning knobs to shape the closed-loop response when using either the multivariable or the distributed controller are the cost function weights on the penalty of the manipulated variables (gate positions, in the case of this paper). In the multivariable case, these weights are defined by the weight matrix \( R = \text{diag}(\rho_1, \rho_2, \rho_3) \). In the distributed case, the cost function associated to each canal subsystem \( i \) depends on the scalar weight \( R_i = \rho_i \). For a given local controller \( C_i \), increasing the weight \( \rho_i \) slightly reduces the overshoot and reduces the closed-loop bandwidth, resulting in a less aggressive controller, as seen in Fig. 5. The bandwidth reduction has the advantage of increasing robustness with respect to high-frequency plant unmodeled dynamics.

Fig. 6 shows the effect on the overshoot of an increase in \( \alpha \). When applied to the canal considered, the convergence of the distributed LQG algorithm depends on the choice of the weights \( \rho_i \) on the penalty of the manipulated variable. Although the iterations do not converge for low values of \( \rho_i \) when all the \( \rho_i \) are equal among themselves, increasing this parameter to a value greater than 10° yields a spectral radius below 0.5 and hence a quite fast convergence. In the experiments reported, the number of iterations performed in each sampling interval is 20.

In experiment 3, the distributed LQG algorithm proposed is tested. Fig. 7 shows results with set-point changes. A comparison with Fig. 3 allows the conclusion that this suboptimal algorithm yields a performance comparable to one of the centralized multivariables.

Fig. 8 shows experimental results on the rejection of disturbances caused by turnouts when distributed LQG is used. Fig. 9 documents the turnout flow for the different pools.

**Quantization of Gate Command**

The PLCs that drive the canal gates include a quantization effect. As seen in Fig. 10, the gate actually moves only when the command differs from the previous gate position by at least \( \delta = 5 \text{ mm} \).
Fig. 5. Simulation results using a nonlinear canal model with the distributed LQG controller for $\alpha = 1$ and different values of the weight $R_2 = \rho_2$; (a) Step responses of pool 2; (b) their effect in pool 1

Although this feature is included as a practical way to prevent excessive actuation, which may result in motor damage, it causes difficulties to the controller.

As shown in the appendix, if the controller is designed such that the ideal quantization-free closed loop is asymptotically stable, then the closed-loop system with quantization is stable and verifies the bound

$$|\tilde{y}(k)| < \delta M$$  \hspace{1cm} (24)

where $M$ = finite constant; and $\tilde{y}(k)$ = difference between the system output with quantization, $y(k)$, and the system output without quantization, $\tilde{y}(k)$.

Although the system with quantization is stable, as concluded from Eq. (24), it is not asymptotically stable. In other words, if $k$ increases without bound, the output does not converge to the reference but only approaches it with an error that may not vanish. This can be understood in qualitative terms by considering the situation in which the output is so close to the reference that the elimination of the tracking error requires an adjustment of the gate position.

Fig. 6. Simulation results of the closed-loop step response using a nonlinear canal model with the distributed LQG controller for $\alpha = 1$ and for $\alpha = 1.004$. 

860 / JOURNAL OF IRRIGATION AND DRAINAGE ENGINEERING © ASCE / OCTOBER 2013
smaller than $\delta$. If the controller has integral effect, it will generate a command that will tend to eliminate the error and will grow without affecting the physical plant until a change of $\delta$ occurs. Because this change is in excess, the sign of the tracking error will change and the controller will then try to move the actuator in the opposite direction.

The previous sequence of events will be repeated, leading to the occurrence of an oscillation of small amplitude that limits the precision of reference tracking, as shown in Fig. 11 (experiment 5). Fig. 12 represents the same data in the plane defined by the variable's gate position and downstream pool water level to show that a closed curve is obtained, corresponding to a limit cycle.

The amplitudes of oscillations induced by quantization in the manipulated variable can be computed using the describing function method as done when studying the impact of numeric quantization in digital control (Franklin et al. 1998) or some switched circuits in power electronics (Peterchev and Sanders 2003). It is very difficult to eliminate the oscillations induced by quantization (Franklin et al. 1998; Peterchev and Sanders 2003). Changing the controller to avoid the oscillations implies a drastic reduction of the gain in the frequency range where the loop gain has a phase of $-\pi$, to prevent

Fig. 7. Experiment 3: Distributed LQG control; experimental results with set-point changes

Fig. 8. Experiment 4: Distributed LQG control; experimental results on the rejection of disturbances caused by turnouts

Fig. 9. Experiment 4: Distributed LQG control. Turnout flows in the experiment reported in Fig. 8
the crossing between the locus of the loop gain and the describing function. This approach has the drawback of reducing the loop bandwidth and implying slower response to both disturbances and setpoint changes. Another possibility is to inject a dither signal that disturbs the sequence of events that result in the oscillation and stopping it. As remarked in Franklin et al. (1998), however, it is very difficult to provide general guidelines for dither design.

When controlling a water delivery canal, it is observed that certain disturbances or change in operating conditions act like a dither signal and stop the oscillation. In any case, the final tracking error may only vanish by chance, being in general within the bounds established by Eq. (24). From a practical point of view, one should tune the controller as well as possible by adjusting its bandwidth in order not to be overreactive to high-frequency disturbances and noise. Only then should the quantization be included, with a value of δ adjusted to be as small as possible while limiting excess of frequent motor moves.

Conclusions

A distributed LQG control structure has been developed and demonstrated on close upstream water level control in a water delivery canal. The distributed controller relies on a coordination algorithm between control agents of neighbor gates. This algorithm consists of an iterative procedure obtained using game theory concepts and explores LQG design formulas for optimization in the presence of accessible disturbances, assuming the approximation that interaction between canal pool subsystem models is due only through the manipulated variables.

The comparison with centralized multivariable control shows that the suboptimal distributed LQG algorithm yields a close approximation to its performance. Significant practical aspects consist of the inclusion of a nonlinear filter to prevent fast variations of the measured output and the inclusion of a prefilter to smooth reference changes and the modification of the quadratic cost with exponential weights to constrain the position of closed-loop poles such as to increase the stability margin.

Adjusting the values of the weights ρ, i = 1, . . . , 4, of the penalty in the manipulated variable is an important tuning knob. Increasing the weights reduces the controller bandwidth and gain and improves robustness with respect to plant high-frequency unmodeled dynamics.

The existence of a quantization effect embedded by software in the manipulated variables as a practical way to prevent the excessive actuation of the motors driving the gates actually limits the tracking precision of the controller and may led to the occurrence of an oscillation.

Appendix: Stability with Gate Position Quantization

The stability of the closed-loop system when a quantizing effect is included in the gate position can be established using methods known for digital control (Franklin et al. 1998). Consider thus the closed-loop control system with a quantization effect in the gate position. This is equivalent to a linear control system in which a disturbance ε(k) is added to the manipulated plant input that is given by the difference between the value of the manipulated variable computed by the controller and the quantized one, actually applied to the gate command. For all time instants k, this disturbance is bounded by

$$|\varepsilon(k)| \leq \delta$$  (25)
where $\delta$ = quantization interval. In closed loop, the system with quantization can thus be represented by the equivalent linear system

$$Y(z) = H(z)R(z) - H_d(z)E(z)$$  \hspace{1cm} (26)

where $E(z), R(z),$ and $Y(z)$ are the Z transforms of $e(t), r(t),$ and of the output to control $y,$ respectively. The symbols $H(z)$ and $H_d(z)$ denote the discrete transfer functions between the reference and the output and between the disturbance and the output. It is assumed that, as when using LQG, the controller is designed such that $H(z)$ has all its poles on the left-hand side of the complex plane, and hence is bounded in, bounded out (BIBO) stable (Franklin et al. 1998).

Let $\hat{y}$ be the output of the ideal quantization free closed-loop system and $\tilde{Y}$ its Z transform. The error between the actual closed-loop model output (that includes the quantization) and the ideal quantization free closed-loop model output is

$$\hat{y}(k) = \tilde{y}(k) - y(k)$$  \hspace{1cm} (27)

and its Z transform $\tilde{Y}$ satisfies

$$\tilde{Y}(z) = H_d(z)E(z)$$  \hspace{1cm} (28)

or in the time domain

$$\hat{y}(k) = \sum_{k=0}^{n} h_d(k)e(n-k)$$  \hspace{1cm} (29)

where $n$ = degree of $H(z)$; and $h$ = its impulse response.

Take the modulus of both sides of Eq. (29), use the bound Eq. (25), and add a positive quantity to the right-hand side to conclude that

$$|\hat{y}(k)| \leq \delta \sum_{k=0}^{\infty} |h_d(k)|$$  \hspace{1cm} (30)

Because $H(z)$ is BIBO stable, there exists a finite constant $M$ such that

$$\sum_{k=0}^{\infty} |h_d(k)|M$$  \hspace{1cm} (31)

Combining Eq. (30) with Eq. (31) yields Eq. (24).

Acknowledgments

This work has been supported by FCT, Portugal, under contracts PTDC/EEA-CRO/102102/2008, PTDC/EMS-CRO/2042/2012 and PEst-OE/EEI/LA0021/2013.

References


