Reconfigurable distributed control of water delivery canals

João M. Lemos
INESC-ID/IST/UTL
R. Alves Redol 9
1000-029, Lisboa - Portugal
Email: jlml@inesc-id.pt

Inês Sampaio
INESC-ID
R. Alves Redol 9
1000-029, Lisboa - Portugal
Email: ines.sampaio@ist.utl.pt

Abstract—This work addresses reconfigurable distributed LQG control of multipurpose open water channels, used both for water delivery and transportation. The use of a local control agent structure of the so called local upstream control type ensures that the water level is kept close to a desired level, even in the presence of disturbances caused by water turnout at side offtakes, so as to ensure navigability. Both actuator and sensor faults are addressed. The paper shows how, in the presence of a fault, the controller/sensor network might be reconfigured in order to mitigate its effects. Experimental results in a large scale pilot canal are included to demonstrate the control concepts proposed.

I. INTRODUCTION

A. Motivation

Several open water channels may be used both as water delivery channels, e.g. for irrigation, and as a navigation mean. Depending on the operational objectives, different control structures may be employed on these systems. While on a water delivery canal dedicated to irrigation the main purpose might be to save water, implying the use of distant upstream control or of its variants [1], other choices are preferred if the major aim is to ensure a desired water level to allow navigation and simultaneously use part of the water for irrigation. In this last case, local upstream control [1] might be an option since it yields a fast rejection of disturbances induced by water turnout at side offtakes.

On the other way, the complexity of large scale water channel systems [8], together with requirements on reliability and quality of service specifications, provide a strong motivation to consider fault tolerant control methods for this type of systems. The idea consists in exploring the redundancy in their sensors and actuators to reconfigure the control system such as to allow the plant operation to continue, perhaps with some graceful degradation, when a sensor or actuator fails. Furthermore, instead of using a centralized controller, where a single processing unit receives all the information and sends the commands to all the actuators, there is a growing trend to consider a network of distributed control agents that negotiate among themselves in order to reach a consensus on the value of the different manipulated variables.

B. Literature review

The concept of fault tolerant control (FTC) has been the subject of intense research in the last twenty years [2], [3], [4]. This activity yielded a rich bibliography that, of course, cannot be covered here and that comprises aspects such as fault detection and isolation and fault tolerant control design. In what concerns distributed control an important concept is “integrity”, namely the capacity of the system to continue in operation when some part of it fails [5]. Other type of approach models the failures as disturbances that are estimated and compensated by the controller [6].

In what concerns water delivery canal systems a topic that receives attention due to their immediate economic impact related to water saving is leak detection [7]. Other aspects found in the literature are control loop monitoring [12], and reconfiguration to mitigate fault effects [11], which is the issue considered in this work.

Reconfiguring the controller in face of a plant fault falls in the realm of hybrid systems and raises issues related to stability that must be taken into account [9].

C. Contributions and paper structure

The contribution of this paper consists in the presentation of reconfigurable distributed LQG controllers that are tolerant to actuator and sensor faults in a water delivery canal. Experimental results in a pilot canal are included.

The paper is organized as follows: After the introduction in which the work is motivated, a short literature review is made and the main contributions are presented, the problem is formulated in section II, including the description of the canal used for testing and the class of faults considered. Section III describes the distributed LQG controller proposed, including the structure of linearized plant models, the design of local controllers and the coordination algorithm. Section IV considers controller reconfiguration to address actuator and sensor faults. Finally, section V draws conclusions.

II. PROBLEM FORMULATION

A. Canal description

The work reported in this paper was performed at the experimental canal of Núcleo de Hidráulica e Controlo de Canais (Universidade de Évora, Portugal), described in [10].
The canal has four pools with a length of 35m, separated by three undershoot gates, with the last pool ended by an overshoot gate. In this work only the first three gates are used. The maximum nominal design flow is 0.09 m$^3$s$^{-1}$. There are water off-takes downstream from each branch that are orifices in the channel walls, that are used to generate disturbances corresponding to water usage.

Water level sensors are installed downstream of each pool. The water level sensors allow to measure values between 0 mm and 900 mm, a value that corresponds to the canal bank. For pool number $i$, $i = 1, \ldots, 4$, the downstream level is denoted $y_i$ and the opening of gate $i$ is denoted $u_i$. Pool number $i$ ends with gate number $i$.

Each of the actual gate positions $u_{r,i}$, $i = 1, 2, 3$ are manipulated by a command signal $u_i$. However, the PLCs that command gate motors are programmed such that $u_{r,i}$ only moves in response to $u_i$ if $|u_i - u_{r,i}| \geq 0.5$ mm. This dead zone nonlinearity has two types of implications. First of all, they limit the controller achievable precision when tracking a reference and they can even induce small amplitude oscillations. Furthermore, when comparing the signals $u_i$ and $u_{r,i}$ in order to detect a fault, this difference must be taken into consideration.

Following [8], in order to compensate for a nonlinearity, instead of using as manipulated variable the gate positions $u_{r,i}$, the corresponding water flows $q_i$ crossing the gates are used. These are related by

$$q_i = C_{ds}Wu_{r,i}\sqrt{2gh_{upstream,i} - h_{downstream,i}}, \quad (1)$$

where $C_{ds}$ is the discharge coefficient, $W$ is the gate width, $g = 9, 8$m/s is the gravity acceleration, $h_{upstream,i}$ is the water level immediately upstream of the gate and $h_{downstream,i}$ is the water level immediately downstream of the gate. This approach corresponds to represent the canal by a Hammerstein model and to compensate the input nonlinearity using its inverse. The linear controller computes the flow crossing the gates, that is considered to be a virtual command variable $v_i$ and the corresponding gate position is then computed using (1). The discharge coefficient is not estimated separately, but instead is considered to be incorporated in the static gain of the linear plant model.

### III. DISTRIBUTED LQG CONTROL

In distributed control the plant model is assumed to be represented by a graph (see Figure 1) with nodes associated to physical agents that have bonds among them that reflect physical interdependencies and are associated to edges. Correspondingly, the distributed controller is described by a graph whose edges are local control agents and whose edges are associated to communication interconnection links. The distributed controller network and the graph network that represents the plant are interconnected by sensors and actuators. A local control agent (or control agent for short) is a software entity that has the following three components:

- A feedback controller that, given the data from the sensors and the local decision unit, computes the signal commands to the actuators;
- A communication unit, that is able to interchange data with other (neighbor) control agents; this data is made of sensor measures and of (possibly potential) decisions about the local manipulated variable.
- A local decision unit, provided with a coordination algorithm, that is able to modify the value of the manipulated variables computed by the local controller on the basis of pure local feedback.

Each of these elements is described hereafter in this section for the water channel considered.

#### A. Linearized plant model

1) **Overall model:** Around an equilibrium point, the plant is represented by linear state-space models written as

$$x(k+1) = Ax(k) + Bu(k), \quad (2)$$

$$y(k) = Cx(k) \quad (3)$$

where $k \in \mathbb{N}$ denotes discrete time, $x \in \mathbb{R}^n$ is the full canal state, $y \in \mathbb{R}^p$ is the output, with $p$ the number of outputs, $u \in \mathbb{R}^p$ is the manipulated variable and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 3}$ and $C \in \mathbb{R}^{3 \times n}$ are matrices. Assuming operation around a constant equilibrium point, these matrices are identified by constraining the model to have the following structure

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix},$$

2) **Fault description**

The faults to be considered consist in either isolated manipulated variable or sensor faults. Manipulated variable faults consist of the blockade of gate number 2. When a fault occurs, the position of gate 2 remains constant and does not change anymore in response to the corresponding command signal. Sensor faults consist of the signal issued by the $y_2$ becoming stuck at its maximum value. It is assumed that no multiple faults occur. Furthermore, since the focus of the work is on controller reconfiguration and not on fault isolation, we consider the simplified situation in which the fault type is a priori known.
These matrices have dimensions that match the state $x_i$ associated to each pool such that $x = [x_1^T, x_2^T, x_3^T]^T$.

In the presence of a fault, the matrices have the structure

$$A = \begin{bmatrix} A_{F11}^F & 0 \\ 0 & A_{F33}^F \end{bmatrix}, \quad B = \begin{bmatrix} B_{F11}^F & B_{F13}^F \\ B_{F31}^F & B_{F22}^F \end{bmatrix},$$

$$C = \begin{bmatrix} C_{F1}^F & 0 \\ 0 & C_{F3}^F \end{bmatrix}.$$ (5)

The superscript $F$ enhances the fact that the matrix blocks are estimated assuming that a fault has occurred and that they are different from the ones in (4). This structure is imposed to reflect the decomposition of the canal model in submodels, each associated to a different pool. Furthermore, it is assumed that each pool interacts directly only with its neighbors, and only through the input.

2) Local models: From the global multivariable model (2, 3), each pool $i$ is represented by the state model with accessible disturbance $d_i$:

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}v_i(k) + d_i(k)$$ (6)

where

$$d_1(k) = B_{12}v_2(k),$$ (7)

$$d_2(k) = B_{21}v_1(k) + B_{23}v_3(k),$$ (8)

and

$$d_3(k) = B_{32}v_2(k).$$ (9)

B. Control agents

Augmenting each local model $S_i$ with a parallel integrator yields the augmented space-state local model

$$\bar{x}_i(k+1) = \bar{A}_{ii}\bar{x}_i(k) + \bar{B}_{ii}v_i(k)$$ (10)

$$y = \bar{C}\bar{x}(k),$$ (11)

where $\bar{x}_i$ is the augmented state of the local model, given by

$$\bar{x}(k) = \begin{bmatrix} x_i(k) \\ x_{i,i}(k) \end{bmatrix}$$ (12)

with $x_{i,i}$ the state of the integrator added to the local model $S_i$ and

$$\bar{A}_{ii} = \begin{bmatrix} A_{ii} & 0 \\ -T_{x}C_{i} \end{bmatrix}, \quad \bar{B}_{ii} = \begin{bmatrix} B_{ii} \\ 0 \end{bmatrix}, \quad \bar{C}_{i} = \begin{bmatrix} C_{i} & 0 \end{bmatrix}.$$ (13)

Since this realization is not observable, only $x_i$ is estimated, using the estimator that depends only on variables available locally for control agent $C_i$

$$\hat{x}_i(k|k-1) = A_{ii}\hat{x}_i(k-1|k-1) + B_{ii}u_i(k-1),$$ (14)

$$\hat{x}_i(k|k) = \hat{x}_i(k|k-1) + M_{p,i}[y_i(k) - C_i\hat{x}_i(k|k-1)],$$ (15)

where the $M_{p,i}$ are kalman gains, that depend on the solution of an algebraic Riccati equation. The manipulated variable is then given by

$$v_i(k) = -K_x,i\hat{x}_i(k|k) - K_{I,i}x_{i,i}(k),$$ (16)

where

$$K_i = [K_{x,i} K_{I,i}].$$ (17)

is obtained by solving a LQ problem that consists of minimizing the steady state quadratic cost

$$J_i = \sum_{k=1}^{\infty} x_i^T(k)Q_i\bar{x}_i(k) + \bar{u}_i^T(k)\rho_i\bar{u}_i(k),$$ (18)

with

$$Q_i = \begin{bmatrix} C_{i}^T C_{i} & 0 \\ 0 & 1 \end{bmatrix},$$ (19)

and $\rho_i > 0$ a design parameter, and is given by the expression

$$K_i = (I + \frac{1}{\rho_i}\bar{B}_{ii}^T P_i \bar{B}_{ii})^{-1} \frac{1}{\rho_i} \bar{B}_{ii}^T P_i \bar{A}_{ii},$$ (20)

in which $P_i$ is the positive definite solution of the algebraic Riccati equation

$$P_i = \bar{A}_{ii}^T P_i (I + \bar{B}_{ii}^T P_i \bar{B}_{ii})^{-1} \bar{A}_{ii} + Q_i.$$ (21)

C. Coordination algorithm

When using distributed control, each gate is manipulated by a SISO controller that selects its moves so as to drive the corresponding water level to the reference value. In addition, there is a correction to achieve a coordinated action. The coordination among controllers is performed by the following algorithm:

Coordination algorithm
At the beginning of each sample time, compute $v_{i,0}(k)$ by solving a LQG problem associated to model (6) and assuming $d_{i}(k) = 0$.

For $j = 1$ up to $j = N_c$ recursively perform the following cycle:
1) For $i = 1, 2, 3$, compute $d_{i,j-1}(k)$ using (7-9) with $v_{i}(k)$ replaced by $v_{i,j-1}(k)$;
2) For $i = 1, 2, 3$, compute $v_{i,j}(k)$ by solving a LQ problem associated to model (6);
Apply to the plant the control given by

$$v_i(k) = v_{i,j}(k)$$ (22)
IV. Controller reconfiguration

In response to a fault that inhibits the action of one of the local control agents (either missing communication links or a sensor or actuator fault) the strategy proposed consists in the reconfiguration of the distributed controller network that consists in the activation of new communication links, together with the redefinition of the control objectives that will probably suffer some degradation.

Assume, as an example, the situation shown in Figure 2 in which the actuator associated to the local control agent CA3 fails. After the moment in which this fault is detected, the local agents CA2 and CA4, that are neighbors to CA3, no longer negotiate with this agent, but start negotiating directly among themselves the value of their respective manipulated variables. Furthermore, the control objectives that depend directly on CA3 may no longer be attainable. For instance, if the plant is a water channel and the actuators associated to the control agents are gates, it is no longer possible to control with precision the level associated to the gate that corresponds to CA3.

A. Actuator faults

1) Reconfiguration mechanism: Figure 3 shows a discrete state diagram that explains how controller reconfiguration is performed when an actuator fault occurs in the water channel considered in this paper. For simplicity, only the occurrence of faults in gate 2 are considered. State $S_1$ corresponds to the situation in which all gates are working normally with a controller $C_N$ that matches this situation. When a fault occurs, the system state switches to $S_2$, in which gate 2 is faulty (blocked) but the controller used is still the one designed for the no fault situation.

When the fault is detected, the state switches to $S_3$, in which a controller $C_F$ designed for the faulty situation is connected to the canal. When the fault is recovered (gate 2 returns to normal operation), the state returns to $S_1$. A dwell time condition is imposed to avoid instability that might arise due to fast switching [13]. This means that, once a controller is applied to the plant, it will remain so for at least a minimum time period (called dwell time).

When distributed control is used, the controller designed for normal operation (shown in figure 4), $C_N$, consists of 3 SISO LQG controllers $C_1$, $C_2$ and $C_3$, each regulating a pool and such that each individual controller negotiates the control variable with its neighbors. This means that, in states $S_1$ and $S_2$, $C_1$ negotiates with $C_2$, $C_2$ negotiates with $C_1$ and with $C_3$ and $C_3$ negotiates with $C_2$. The controller for the faulty condition (shown in figure 5) is made just of two SISO controllers that control pools 1 and 3 and negotiate with each other.

A bumpless transfer algorithm is used in order to ensure the continuity of the manipulated variable command applied
to the gates when there are switching among controllers.

For actuator faults, the fault detection algorithm operates as shown in the block diagram of figure 6. For each gate $i$, $i = 1, 2, 3$, define the error $\tilde{u}_i$ between the command of the gate position $u_i$ and the actual gate position $u_{r,i}$,

$$\tilde{u}(k) = u_i(k) - u_{r,i}(k)$$

A performance index $\Pi$ is computed from this error by

$$\Pi(k) = \gamma \Pi(k - 1) + (1 - \gamma) |\tilde{u}(k)|$$

If $\Pi(k) \geq \Pi_{\text{max}}$, where $\Pi_{\text{max}}$ is a given threshold, then it is decided that a fault has occurred.

2) Experimental results: Figure 7 shows results obtained with the distributed LQG controller and reconfiguration. At the time instant marked by a red vertical line, a fault occurs that forces gate 2 to become stuck. Shortly after, at the instant marked by the yellow vertical line, this fault is detected, and the controller is reconfigured as explained. From this moment on, there is no warranty on the value of the level $J_2$, but $J_1$ and $J_3$ continue to be controlled. The effect of coordination is apparent in the setpoint decrease of pool 1, close to time 5300 s. Gate 1 opens to allow the water in pool 1 to be drained, but gate 3 also opens despite the water level at gate 3 remains very little disturbed. Therefore, the opening of gate 3 is not due to a feedback effect. Instead, it is due to the coordination between controllers 1 and 3.

B. Sensor faults

1) Reconfiguration mechanism: The sensor fault considered consists in the sensor of level $J_2$ to become constant and equal to the maximum variable of this variable (800 mm). The sensor fault is detected by building an error signal between the measured signal and its estimate computed from the level sensor in the middle of the pool. An indicator variable is then obtained in a similar way as for actuator fault detection. When a sensor fault is detected, the signal used for feedback is the reconstruction obtained from the level sensor in the middle of the pool.

2) Experimental results: Figure 8 shows an example of sensor reconfiguration of a distributed LQG controller applied to the water channel in response to fault in the sensor level of $J_2$. After the first vertical yellow line the sensor fault has been detected (shortly after its occurrence). The second
vertical yellow line marks the end of the fault, in which the sensor returns to the normal condition. When the fault is active, the controller uses as feedback signal a reconstructed signal obtained by filtering the measures from other sensors. Although there is a degradation in control performance in pool 2, the results are still acceptable. In case nothing would have been done to accommodate the fault, gate 2 would open completely because it would “see” a very high level. Furthermore, this would completely mix-up the coordination procedure, leading ultimately to a serious process failure. The pair of yellow lines just after 6000 s corresponds to a false alarm of short duration.

V. CONCLUSION

A reconfigurable distributed controller for tackling faults in water channels is proposed and illustrated experimentally. It explores distribution of control agents and redundancy in communication links and sensors. The formulation as a hybrid system allows a stability analysis, which not made in this paper.

ACKNOWLEDGMENT

This article was supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, Portugal, under contracts PTDC/EMS-CRO/2042/2012 (project ORCHESTRA – Distributed Optimization and Control of Large Scale Water Delivery Systems) and PEst-EEI/LA0021/2011.

REFERENCES


Fig. 8. Distributed LQG controller. Reconfiguration after a fault in the sensor that measures \( y_2 \). Pool levels (above), gate positions (middle) and reconstructed variables (below).