

Distributed Instruments for Control and Diagnosis Applied to a Water Distribution System

Didier GEORGES (1), Eric BENOIT(2), André CHOVIN(3), Damien KOENIG(1),
Benoit MARX(1), Gilles MAURIS(2)

(1) Equipe GSYS - ESISAR,
Laboratoire d'Automatique de Gre-
noble
UMR 5528 INPG - CNRS - UJF,
BP 54, 26902 Valence cedex 9,
France
Tel: +33 475 759 409,
Fax: +33 475 435 642,
Email:Didier.Georges@inpg.fr

(2) Laboratoire d'Automatique et de
MicroInformatique Industrielle
LAMII
ESIA - Université de Savoie,
BP 806,74016 Annecy, France
Tel: +33 450 096 540,
Fax: +33 450 096 559,
Email: Eric.Benoit@univ-savoie.fr

(3) CROUZET Automatismes
2 rue du Docteur Abel BP59,
26902 VALENCE Cedex 09,
FRANCE
Tel : +33 475 448 609,
Fax: +33 475 559 803,
Email: achovin@crouzet.com

Abstract – this paper presents the design of intelligent instruments which are distributed on a CANopen fieldbus in order not only to synthesise and implement robust control algorithms but also to perform fault diagnosis on distributed industrial processes. Firstly, we shall shortly present the methodology and the context of the project, secondly the process is described and thirdly the main objectives of the work are exposed.

Keywords – distributed instruments, intelligent instruments, control, diagnosis

I. INTRODUCTION

The work presented in this paper is part of SYCADI program (Solutions pour SYstèmes à Capteurs et Actionneurs Distribués: solutions for distributed sensor and actuator-based systems), a research program supported by Région Rhône-Alpes (France), gathering two research laboratories (LAG and LAMII) and a industrial company: the automation component producer Crouzet Automatismes, part of Schneider Group.

The major goals of this partnership is:

- Development of industrial tools dedicated to the design of smart components,
- Design of monitoring and control methodologies for systems controlled via fieldbus
- Implementation of the resulting algorithms in intelligent sensor-actuator instruments connected to field buses.

The paper is organised as follows: in the second section, the experimental process is presented and its main characteristics are highlighted; in the third section the different objectives are detailed, in particular, attention is focused on robust control law design and system diagnosis and finally implementation of the derived algorithms on distributed instruments linked to a field bus.

II. DESCRIPTION OF THE EXPERIMENTAL PROCESS

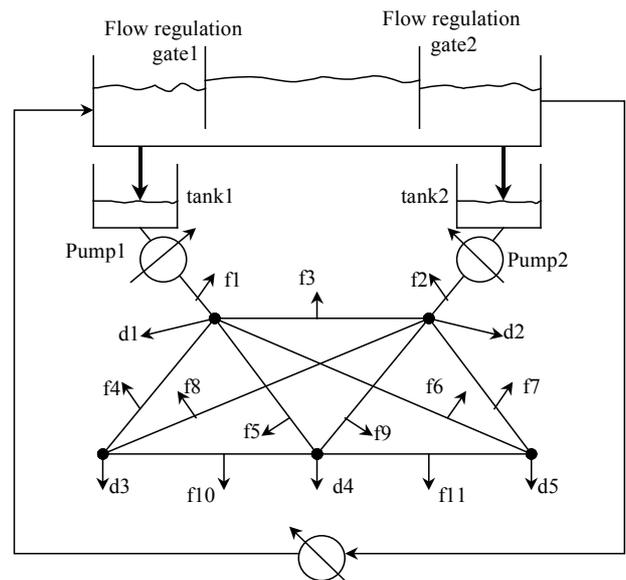


Figure 1 : testbed based on an experimental small-size canal connected to a water distribution network

The process is a technological platform composed of an open-channel hydraulic system connected to a water distribution system (see figure 1). The overall system is designed to simulate a real water distribution system connected to an irrigation canal transporting and providing water to agricultural consumers. All the sensors and actuators used for control and diagnosis are linked together via a CANopen fieldbus

A state space model representing the water distribution network dynamics is established by considering the behavioural laws of the different components (pumps, pipes and nodes) and the dynamical equations arising from flow conservation principle applied to each tank. The former equations can be written as solutions of non linear algebraic relations $g_1(h,q,u,d,f)=0$, where h is the pressure vector, q the flow vector, u the control input vector, d the disturbance and f the failures, and the latter can be written as dynamical equations $dh_i/dt=g_2(h,q,u,d,f)$ where h_i are water level in the tanks. It must be noticed that only two relations out of eighteen are some dynamical equations: in fact, a water distribution network is an algebraic-differential or singular system. This class of systems has received a considerable amount of attention in the last two decades (see [1][2][3]) but results concerning diagnosis and control are not so numerous.

III. RESEARCH OBJECTIVES

The overall research goal is divided into three main tasks:

- Decentralized control of the irrigation canal
- Diagnosis and leakage detection of the water distribution network
- Intelligent instrument design and diagnosis and control algorithm implementation for system monitoring

1. Decentralized control of the irrigation canal

Dynamics of open-channel hydraulic systems, such as rivers or irrigation canals, are governed by the well-known Saint-Venant nonlinear partial differential equations (see [4][5]):

$$B \frac{\partial h(x,t)}{\partial t} + \frac{\partial Q(x,t)}{\partial x} = q(t) \quad (1)$$

$$\frac{\partial Q(x,t)}{\partial t} + \frac{1}{B} \frac{\partial (Q(x,t)^2 / h(x,t))}{\partial x} + gBh(x,t) \left(\frac{\partial h(x,t)}{\partial x} - I + J(x,t) \right) = k_q q(t) V(x,t) \quad (2)$$

Where $h(x,t)$ and $Q(x,t)$ represent the water level and the water flow rate in the canal, respectively. These variables are defined as function of both time and space coordinate x along the canal. $q(t)$ is the distributed inflow or withdrawal along the canal. I and J represent the canal slope and the friction respectively. B is the canal width. Irrigation canals are known to be large-scale complex nonlinear systems. For that reason, effective control of such systems relies on the use of linear or nonlinear control techniques in order to regulate water level in the goal of both water demand satisfaction and flood avoidance. Real-time implementation of such control designs relies on the use of distributed sensors and actuators together

with distributed implementation of the control algorithms interconnected through a field bus [7].

In order to obtain a suitable control model, system reduction techniques have been proposed in order to get a reduced-order model, that can be tractable for control purposes. In this work, we are also concerned with optimal control architecture design: that is optimal field bus architecture design together with optimal location of the distributed water level sensors and regulator gates, with the goal of increasing both observability and controllability of the controlled system [6].

2. Diagnosis and leakage detection of the water distribution network

The system is affected by failures (pipe leakage), thus diagnosis is performed to detect and isolate these failures. Roughly speaking, detection is to notice the occurrence of one or several fault and isolation is the precise knowledge of which fault has occurred.

The method used to solve this fault detection and isolation (FDI) problem is based on diagnosis techniques using structural residual generation. Taking into consideration the knowledge of the system we can compute a comparison of the measured evolution of the system and its nominal behaviour (without any failure). Then residual signals $r(t)$ are generated such that $r(t)$ is close to zero when no failure occur and $r(t)$ is significantly different to zero in the opposite case. There exist two main methods using balance equation for residual generation, the *a posteriori* methods using state observer to compare measurements and estimations and *a priori* methods using un-filtered data to derive residual signals.

The former method need to treat all the available information, in other words all the measurements must be computed in a single algorithm. On the opposite, with the latter one [7], a residual signal can be associated to each component of the network. The algorithm used to compute the residual associated to each component only needs measurements of adjacent components. Thus we have chosen the second one. This distributed aspect of the residual generation motivates the use of intelligent instruments and fieldbus.

Firstly, we represent the system, only considering its structure. Thus we model the system with a directed graph, each hydraulic component (pump, node, tank,...) becomes a node and each arc corresponds to one flow. A node has to be added to model the exosystem (source of the inputs flows of the tanks and target of the leaks f and the demands d on fig. 1). Then an incidence matrix $M=(m_{ij}) \in N^{n \times m}$ is associated with the graph by

$$m_{ij} = \begin{cases} 1 & \text{if the node } i \text{ is the target of the arc } j \\ -1 & \text{if the node } i \text{ is the source of the arc } j \\ 0 & \text{else} \end{cases} \quad (3)$$

Secondly, using this matrix representation we generate residual signals. The balance equation of the nodes can be expressed by $M_i \cdot q = 0$, where M_i denotes the i^{th} row of M . Thus if a fault affecting one of the arcs connected to the node i occurs, $r_i = M_i \cdot q$ becomes non null, in other words it is a residual which can be used for the detection of this fault.

Moreover, let assume that measurements are submitted to errors of measurements e , with e characterised by a centred normal law of variance V , i.e. $e \sim N(0, V)$. Using the matrix variance we can define the normalised residual signals r_{ni} associated with the node i

$$r_i = M_i \cdot q$$

$$r_{ni} = \frac{M_i \cdot q}{\sqrt{(MVM^T)_{(i,i)}}} \quad \text{for } i = 1, \dots, n \quad (4)$$

The residual r_i is sensitive to all the faults affecting one of the input or output flows of node i , thus fault detection is possible, but due to the interconnections these residuals are not sufficient for fault isolation. Other residuals are required. We can use the same method to associate a residual signal with each arc of the directed graph. Therefore we define r_{ai} (resp. r_{nai}) the residual (resp. normalised residual) corresponding to the balance equation of the arc i by

$$r_{ai} = (\text{abs}(M^T)M)_{i,q}$$

$$r_{nai} = \frac{(\text{abs}(M^T)M)_{i,q}}{\sqrt{((\text{abs}(M^T)M)V(\text{abs}(M^T)M)^T)_{(i,i)}}} \quad \text{for } i = 1, \dots, n \quad (5)$$

Let note that the residual r_{ai} is sensitive to all the faults affecting the input and output flows of the target node and source node of the arc i , but it is non sensitive to a fault affecting the arc i itself.

Thirdly, to detect and isolate the fault, we set decision rules to define the alarms associated with each arc. These rules use boolean variables associated with the residual (of an arc or a node), those variables are set to 1 if and only if a fixed threshold is passed

$$b_i = \begin{cases} 1 & \text{if } |r_{ni}| \geq \epsilon_i \\ 0 & \text{if } |r_{ni}| < \epsilon_i \end{cases} \quad (6)$$

An elementary decision rule to detect a fault affecting the arc i is given by

$$A(a_i) = b_{I(a_i)} \cdot b_{O(a_i)} \cdot \overline{b_{a_i}} \quad (7)$$

where the arc a_i inputs the node $I(a_i)$ and outputs the node $O(a_i)$. this decision rule can be used in the case of unique fault affecting the system, but if several faults occur simultaneously the boolean b_{ai} may remain equal to 1 and then $A(a_i)$

may become insensitive. Thus, a more complex rule must be set to take into account the multiple defaults

$$A(a_i) = b_{I(a_i)} \cdot b_{O(a_i)} \cdot \overline{b_{a_i}} + b_{I(a_i)} \cdot b_{O(a_i)} \cdot b_{a_i} \left(\prod_{a_k \in IS, IT} b_{I(a_k)} \cdot b_{a_k} \right) \left(\prod_{a_l \in OS, OT} b_{O(a_l)} \cdot b_{a_l} \right) \quad (8)$$

where IS (resp. IT) denotes the set of arcs inputting the source (resp. target) node of a_i and OS (resp. OT) denotes the set of arcs outputting the source (resp. target) node of a_i .

It is important to notice, that the alarm $A(a_i)$ only requires the knowledge of the adjacent measurements, thus this methodology is particularly adapted to distributed instrumentation.

3. Intelligent instrument design and control algorithm implementation for system monitoring

The design of intelligent instruments requires more and more multiple competences. The design complexity of instruments requires several specialists such as, sensor or actuator designers, and software designer. Therefore it makes it difficult to guarantee a safe design. The model approach [8], presented in this paper, splits the instrument design into three levels:

- The instrument user specification level based on an external model of the instrument.
- The instrument designer level based on an internal model of the instrument.
- The software designer level where atomic pieces of software are created.

It also unifies these levels and allows a list of model verifications that improve the safety of the design. Using an internal model and an external model increase the safety of the instrument design and the instrument use. The separation between the designer action and the software designer work allows to quickly adapt a product to a specific kind of user. Indeed, the software designer create basic algorithms and pieces of software dealing with the instrument hardware, and the instrument designer just have to create the instrument behavior in order to respond to the user needs.

The external instrument model is based on User Operating Modes (USOM) [9]. In this model, instruments functionalities are considered as services, and instruments states called USOM are defined as subsets of instruments services. In this application, the set of possible modes is intentionally restricted to the modes defined by the CANOpen field bus protocol, i.e. Pre-operational, Operational and Stopped modes [10]. Consider now an internal point of view of the instrument. Preceding services are then called external services. New services, called internal services, are defined. These internal services are invisible by the user, and they can only be requested by an other service of this sensor. They are in-

cluding algorithms like filtering, signal processing or are dealing with the hardware of the instrument such as transducers or field bus.

In the water distribution system, the instruments used on the open-channel irrigation canal are localized near the water gates. Each one takes pressure measurement from a Pitot tube, and modify the gate position with a brushless motor.

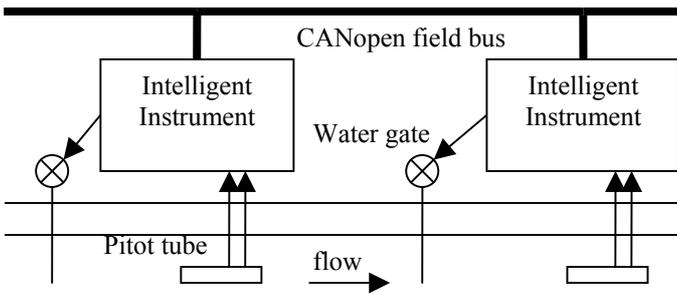


Figure 2: Open-channel irrigation canal system and some Intelligent instruments.

The software designer defines 8 internal services for this kind of instrument. 2 internal services acquire pressure values from the Pitot tube. One computes the flow speed and the level of the water in the canal. 2 other services send the values of speed and level through the CANOpen fieldbus. A level control and a flow control are performed by 2 different internal services. These one use values of local level and flow, and external values given by another instrument. Finally, an internal service performs a position control of the gate. The dependence graph of internal services is then defined (see Fig. 3).

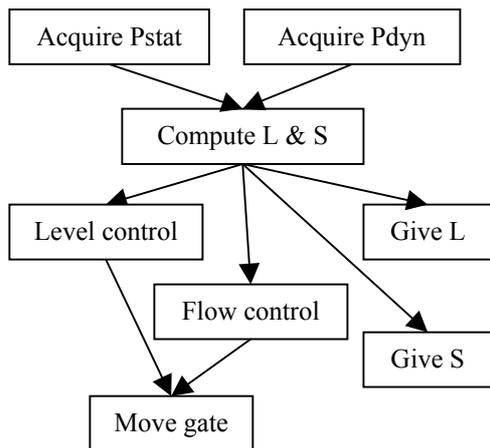


Figure 3: Dependence graph of internal services of an instrument on the Open-channel irrigation canal.

At this step the instrument designer can define instrument external services. Each external service is simply defined as a set of internal services. In the following figure, the *Measure-*

ment service that just sends speed and level values, and the *Flow* service that performs a flow control, are defined as a set of internal services. Note that several services can use the same internal services.

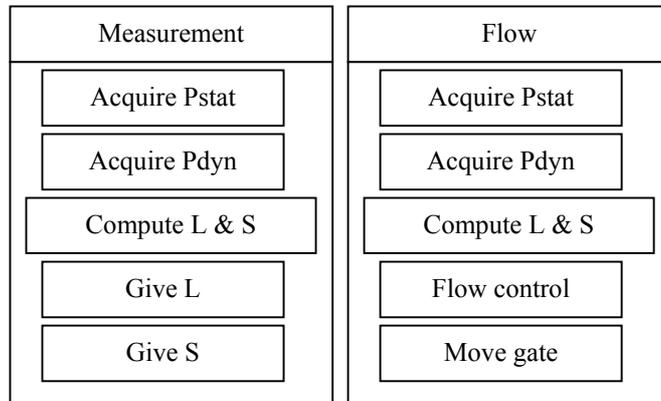


Figure 4: External services definition.

The last step of the instrument design is the generation of the software of each instrument. A set of tool dedicated to the automatic generation of the software have been created during this project. The first tool performs a model-checking in order to verify the consistency of the design. Then it generates a description of the instrument into a well-known language. The language chosen is XML. Indeed, the XML approach is based on the separation between the information and its representation. In the instrument design context, the XML file of an instrument includes the instrument behaviour defined with the internal model and the external model. The software of the instrument, and its documentation can be considered as representations of the instrument.

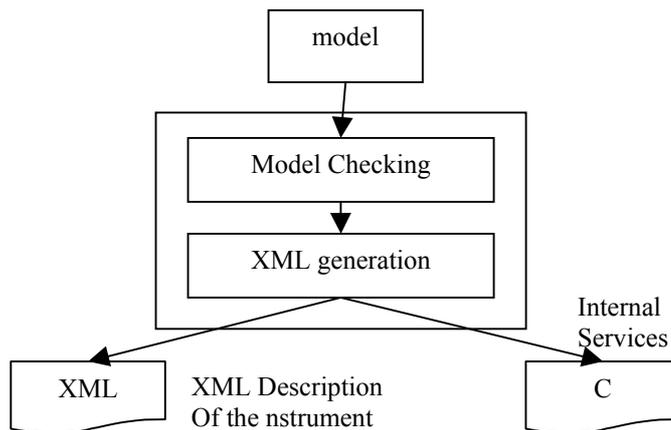


Figure 5: Model checking and XML generation of the instrument description

Then several style sheets, i.e. XSL files, are applied on the XML information file of the instrument in order to generates

a C source-code, an electronic datasheet, and a HTML documentation. Each style sheet defines a representation style that is applied to the XML file in order to produce the desired documents.

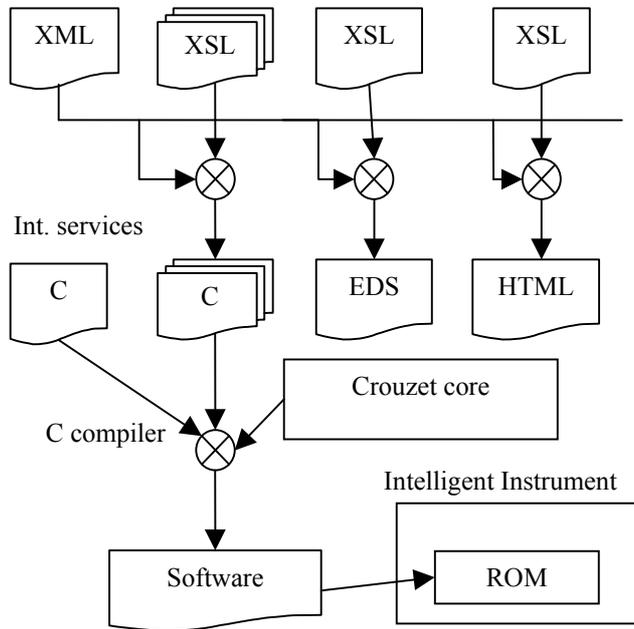


Figure 6 : Software and documents generation.

Finally, all intelligent instruments of this technological platform are designed with this modeling and their embedded software generated with the presented tools. As the behavior of the application can be implemented into the instruments, it is not necessary to use any centralized control system like automation controller. The idea is to distribute the application software into intelligent instruments. This solution is used for the open-channel irrigation canal.

CONCLUSION

The results obtained by the partnership between the two research laboratories (LAG and LAMII) and the industrial

company Crouzet Automatismes demonstrate the possibility of the design of complex distributed applications. It is now possible to create intelligent instruments that include complex functions with a behavior easy to design. The control of the open-channel irrigation canal and the leakage detection by diagnosis of the water distribution network is being tested on the technological platform.

REFERENCES

- [1] D. Cobb, Controllability, Observability and duality in singular systems, IEEE Transactions On Automatic Control, vol. 29, no. 12, pp. 1076-1082, 1984.
- [2] D. Cobb, Descriptor Variable Systems and Optimal State Regulation, IEEE Transactions On Automatic Control, vol. 28, no. 5, pp. 601-611, 1983.
- [3] L. Dai, Singular control systems, Berlin, Springer, 1989.
- [4] D. Georges, J.-F. Dulhoste, G. Besançon, Modelling and control of water flow dynamics via a collocation method, Proc. of MTNS'2000, Perpignan (France), july 2000.
- [5] D. Georges, Nonlinear implicit state-observer design for the monitoring of open-channel hydraulic systems, Proc. of IEEE International Conference on Systems, Man and Cybernetics, October 1995, Vancouver, Canada.
- [6] M. Ghetie, and S. Mehrdad, On-Line Fault Detection and Isolation Using Unbalanced Residuals, Proc. of the 38th Conf. on Decision and Control, Phoenix Arizona, USA, 1999.
- [7] D. Georges, Decentralized adaptive control of water distribution system, Proc. of 3rd IEEE Conference on Control Applications, Glasgow, (Scotland, UK), august 1994.
- [8] J. Tailland, L. Foulloy, E. Benoit, Automatic Generation of Intelligent Instruments from Internal Model, Proc. of 4th IFAC Int. Symp. SICICA'2000, Buenos Aires, Argentina, Sept. 2000, pp 337-342.
- [9] A. Bouras, M. Staroswiecki, Building distributed architectures by the interconnection of intelligent instruments, IFAC INCOM'98 on "Information Control Problems in Manufacturing Technology", Nancy & Metz (France), June 1998.
- [10] E. Benoit, A. Chovin, L. Foulloy, A. Chatenay, G. Mauris, Safe design of CANopen distributed Instruments, Proc. of the IEEE Instrumentation & Measurement technology IMTC/2001 Conference, Vol. 3, Budapest, Hungary, 2001, pp 1768-1772.

ACKNOWLEDGMENT

The authors are grateful to Région Rhône-Alpes for financing this research program.