2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

FrBI02.5

Reconfiguration through a Component Oriented Hybrid Architecture

Ahmed Guadri, Nathalie Dangoumau, Etienne Craye

***Abstract*— In this article, we present a new methodology for the reconfiguration of a hybrid system that relies on component-oriented hybrid architecture. Each component is described through a hybrid behavioural automaton, and is controlled through continuous and discrete entries that are injected by the controller. The latter is used until the objectives specifications are not respected. In this case, a central reconfiguration module updates the controller according to a components description database. This architecture is applied in the case of a system of communicating tanks.**

I. INTRODUCTION

Fault tolerant hybrid control has been investigated in scientific literature through a number of approaches such as hierarchical hybrid control ([4], [3], [7]), global qualitative modelling and abstraction ([2], [1]) or component oriented qualitative modelling ([5] or [8]). These models can be exploited for the control and supervision task by supervisory control([2], [6]), reachability analysis...

A major problem arises in controller synthesis for hybrid controllers is the controller synthesis task complexity. Having in mind this obstacle, we propose a control and reconfiguration architecture that relies in initial models of the controller and the interface. These models could be synthesized by several methodologies (for example : the partition of the continuous space state through natural invariants [2]). The reconfiguration task is triggered when the system fails to met some specified objectives, and is achieved in two steps : finding a qualitative inputs sequence so that the current objective could be met, and finding the discrete and continuous controls that will be implemented in the interface according to the generated qualitative sequence.

We applied in the case of a two communicating tanks (figure 1).

II. THE SYSTEM ARCHITECTURE

A. *The process and control architecture*

We propose a reconfiguration architecture (see figure 2 for the particular case of the two tanks system) composed of three main parts : The hybrid system, the control modules (controller+injector+generator) and the Reconfiguration modules (main reconfiguration module+description database).

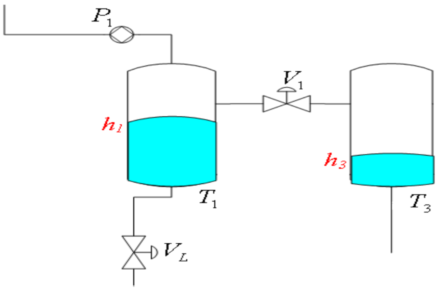

Fig. 1. The three tanks system

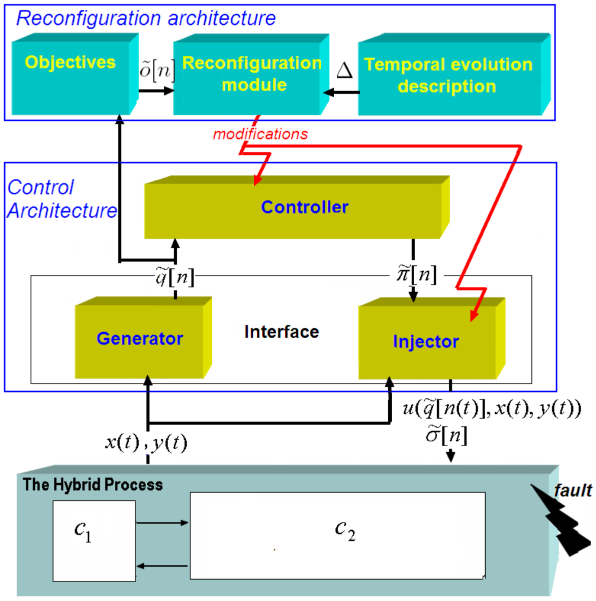

Fig. 2. The global system architecture

This architecture relies in a component oriented beavioral model, where each component dynamics are described through a variant of hybrid behavior automaton and fault automaton (see figure 3 for the particular case of the valve V_L).

For each component, a variant of hybrid automaton specifies the component variables evolution through a set of algebraic differential equations and discrete switches and resets. For example, $V_L.x$ is the V_L internal dynamic continuous variables, $V_L.Y_{read}$ is the read only shared variables, $V_L.Y_{write}$ is the read/write shared variables, $V_L.\Sigma$ is the discrete control events and $V_L.X$ is the continuous state space of the component (An empty state space means that there is no internal continuous dynamical variable).

Ahmed Guadri is a PHD student from the LAGIS laboratory, SED team, Ecole Centrale de Lille, 59650 France `Ahmed.Guadri@ec-lille.fr`

Nathalie Dangoumau is associate professor in Ecole Centrale de Lille, 59650 France `Nathalie.Dangoumau@ec-lille.fr`

Etienne Craye is professor in Ecole Centrale de Lille, 59650 France `Etienne.Craye@ec-lille.fr`

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

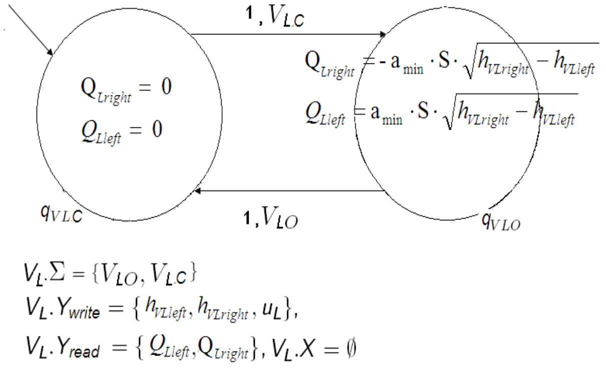

Fig. 3. the behavioral model of V_L

The objectives and the controller are quantified through finite state deterministic automata (figure 4 for the case of the 2 tanks system). The first automaton specifies the prohibited system modes (R_I), and the acceptable temporal succession of system modes, whereas the controller automaton describes the controller mode evolution according to the received symbols from the hybrid process.

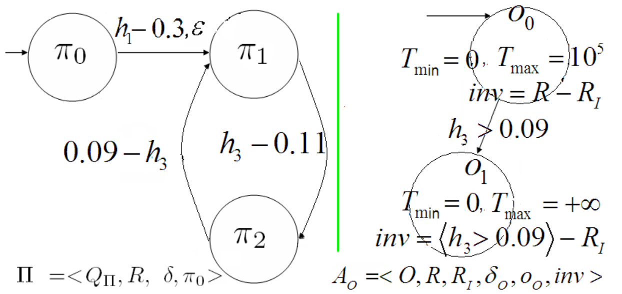

Fig. 4. the objectives and initial controller for the two tanks system

If the controller and the interface are well designed, the hybrid system evolves according to the objectives automaton until a fault causes a failure.

B. The Reconfiguration Architecture

The system design task includes a generation step of a temporal description database of the components evolution. For each component, we specify a function $\Delta(r_1, r_2, r_{neighb}, v, \sigma) = (\Delta^-, \Delta^+)$. This expression is intended to give an upper and lower bound for enabling the evolution from r_1 to r_2 with a control event σ when the environing components are in the modes set r_{neighb}, and when the component receives a continuous input from the $v \in V \in PU$, with PU an initial partition of the input continuous space U.

The update reconfiguration is then :

Algorithm 1: 1) Use the controller and interface as initial control architecture until the system does not meet the objectives automaton. Go to 2

2) Let $\tilde{\pi}[n]$, $\tilde{o}[n]$ be the current controller and objectives state, the reconfiguration module computes a hybrid control sequence Γ in order to reach an invariant of one of the successors of $\tilde{o}[n]$, or to return to the invariant of $\tilde{o}[n]$ itself. This sequence must reach this region within the maximal duration that is specified for the current objective state, which could be achieved thanks to the description database. Go to 3

3) Insert a succession of control states to the controller. The injector is updated in order to associate for each new control state its corresponding generated hybrid control.

The second step of the algorithm 1 updates the controller automaton through a breadth first search of the feasible transitions thanks to the temporal descriptions. As Δ specifies an upper and lower bounds of the transition duration, finding a control sequence such as the sum of its temporal descriptions is within the $[\Delta^-, \Delta^+]$ insures that we could reach the wanted modes within the objective acceptable duration.

For example, suppose that the valve V_1 in the figure 1 is blocked opened. Let n be the step when this fault occurred, with $\tilde{\pi}[n] = \pi[2]$, $\tilde{o}[n] = o_1$. The reconfiguration module performs exhaustive exploration of the possible system evolution according to the temporal evolution database in order to generate a new controller. In practice, this is done by exploitation of the description database in order to find qualitative solutions of the reconfiguration objectives.

III. CONCLUSION AND FUTURE WORK

In this article we have shown an architecture that achieves the reconfiguration of the system continuous and discrete inputs in order to satisfy the objectives specifications. Thereafter, we plan to introduce hierarchy to enable a complex modeling and to exploit available incomplete information for the reconfiguration task.

REFERENCES

[1] Rajeev Alur, *Discrete abstractions of hybrid systems*, Proceedings of the IEEE **88** (2000), no. 7, 971–984.

[2] X.D. Koutsoukos, P. J. Antsaklis, J.A. Stiver, and M.D. Lemmon, *Supervisory control of hybrid systems*, Proceedings of the IEEE **88** (2000), 1026–1049.

[3] John Lygeros, *Hierarchical, hybrid control of large scale systems*, Ph.D. thesis, 1996, A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Engineering.

[4] John Lygeros, Datta N. Godbole, and Mireille broucke, *A fault tolerant control architecture for automated highway systems*, IEEE Transactions on Control Systems Technology **8** (2000), 205–219.

[5] Suzanne Manz and Peter Gohner, *Development of hybrid component models for online monitoring of complex dynamic systems*, vol. 279/2002, ch. 22, pp. 391–418, Springer Berlin/Heidelberg, 2002.

[6] Thomas Moor and Jrg Raisch, *Abstraction based supervisory controller for high order monotone continuous systems*, vol. 279/2002, ch. 14, pp. 246–266, Springer Berlin/Heidelberg, 2002.

[7] Jrg Raisch and Thomas Moor, *Hierarchical hybrid control synthesis and its application to a multiproduct batch plant*, Lecture Notes in Control and Information Sciences **322** (2005), 199–210.

[8] Rong Su, Sherif Abdelwahed, Gabor Karsai, and Gautam Biswas, *Discrete abstractions and supervisory control of switching systems*, IEEE International Conference on Systems, Man and Cybernetics **1** (2003), 415–421.