FAULT-TOLERANT CONTROL OF MULTI-UNIT PROCESS SYSTEMS USING COMMUNICATION NETWORKS [1]

Nael H. El-Farra, Adiwinata Gani and Panagiotis D. Christofides [2]

Department of Chemical Engineering
University of California, Los Angeles, CA 90095-1592

Abstract: This work proposes a methodology for the design of fault-tolerant control systems for chemical plants with distributed interconnected processing units. Bringing together tools from Lyapunov-based nonlinear control and hybrid systems theory, the approach is based on a hierarchical architecture that integrates lower-level feedback control of the individual units with upper-level logic-based supervisory control over communication networks. The local control systems consist each of a family of control configurations connected, via a local communication network, to a local supervisor that orchestrates switching between them on the basis of the stability regions in the event of failures. The local supervisors communicate, through a plant-wide communication network, with a plant supervisor responsible for monitoring the different units and coordinating their responses in a way that minimizes the propagation of failure effects. The communication logic is designed to ensure efficient transmission of information between the units while respecting the inherent limitations in network resources. The proposed approach provides explicit guidelines for managing the various interplays between the tasks of feedback control, switching and communication. The efficacy of the proposed approach is demonstrated through a chemical process example.

Keywords: Hybrid control, Switching logic, Stability regions, Fault-tolerance, Supervisory control, Communication networks, Process systems.

1. INTRODUCTION

Safety and reliability are primary goals in the operation of industrial chemical plants. An important national need currently exists for enhancing the safety and reliability of chemical plants in ways that reduce their vulnerability to serious failures. Increasingly faced with these requirements and other economic drivers, plant operation is relying extensively on highly automated process control systems. Automation, however, tends to increase vulnerability of the plant to faults (e.g., defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops) which, if not appropriately handled in the control system design, can potentially cause a host of undesired economic, environmental, and safety problems that seriously degrade the operating efficiency of the plant. These considerations provide a strong motivation for the development of systematic methods and strategies for the design of fault-tolerant control systems and have motivated many research studies in this area (e.g., see (Willsky, 1998; Yang *et al.*, 1998; Bao *et al.*, 2002) and the references therein).

Given the complex dynamics of chemical processes (due, for example, to the presence of nonlinearities and constraints) and the geographically distributed, interconnected nature of plant units, as well as the large number of distributed sensors and actuators typically involved, the success of any fault-tolerant control strategy requires an integrated approach that brings together several essential elements, including: (1) the design of advanced feedback control algorithms that handle complex dynamics effectively, (2) the design of supervisory switching schemes that orchestrate the transition from the failed control configuration to available well-functioning fall-back configurations to ensure fault-tolerance, and (3) the efficient exchange of information and communication between the different plant units through a high-level supervisor that coordinates the overall plant response in failure situations and minimizes the effects of failure propagation. The realization of such an ap-

[1] Financial support from NSF, CTS-0129571, is gratefully acknowledged.

[2] Corresponding author (e-mail: pdc@seas.ucla.edu)

proach is increasingly aided by a confluence of recent, and ongoing, advances in several areas of process control research, including advances in nonlinear controller designs for constrained chemical processes (e.g., (El-Farra and Christofides, 2001; El-Farra and Christofides, 2003*a*)) and advances in the analysis and control of hybrid process systems (e.g., (Bemporad and Morari, 1999; El-Farra and Christofides, 2002; El-Farra and Christofides, 2003*b*)).

In addition to the above fundamental advances, recent innovations in actuator/sensor and communication technologies are increasingly enabling the integration of communication and control domains. For example, the use of communication networks as media to interconnect the different components in an industrial control system is rapidly increasing and expected to replace the more costly point-to-point connection schemes currently employed in distributed control systems. In addition to the advantage of reduced system wiring, the increased flexibility and ease of maintenance of a system using a network to transfer information is an appealing goal. In the context of fault-tolerant control, systems designed in this manner allow for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when component failure occurs, and in general they allow having a high-level supervisor control over the entire plant. Currently, networked control systems is an active area of research within control engineering (e.g., see (Walsh *et al.*, 2002; Montestruque and Antsaklis, 2003; Tipsuwan and Chow, 2003; Patankar, 2003) and the references therein). The appealing features of communication networks motivate investigating ways for integrating them in the design of fault-tolerant control systems to ensure a timely and coordinated response that minimizes failure propagation effects between plant units.

In a previous work (El-Farra *et al.*, 2004), we presented an approach for fault-tolerant control of single-unit process systems using the idea of integrating feedback and supervisory control over networks. In this paper, we extend our previous work and develop a fault-tolerant control system design methodology, for plants with multiple (distributed) interconnected processing units, that accounts explicitly for the inherent complexities in supervisory control and communication tasks resulting from the distributed interconnected nature of plant units. The approach brings together tools from Lyapunov-based control and hybrid systems theory and is based on a hierarchical distributed architecture that integrates lower-level feedback control of the individual units with upper-level logic-based supervisory control over networks.

The local control systems consist each of a family of feedback control configurations together with a local supervisor that communicates with actuators and sensors, via a local communication network, to orchestrate the transition between the control configurations, on the basis of their fault-recovery regions in the event of failures. The local supervisors communicate, through a plant-wide communication network, with a plant supervisor responsible for monitoring the different units and coordinating their responses in a way that minimizes the propagation of failure effects. The communication logic is designed to ensure efficient transmission of information between the units while respecting the inherent limitations in network resources by minimizing unnecessary network usage and accounting explicitly for the effects of possible delays due to fault-detection, control computations, network communication and actuator activation.

2. PRELIMINARIES

2.1 *System description - problem formulation*

We consider the class of continuous-time, multivariable nonlinear process systems with constraints on the manipulated input, represented by the following state-space description:

$$\begin{aligned} \dot{x}(t) &= f_{k(t)}(x(t)) + \sum_{i=1}^{m} g^i_{k(t)}(x(t))u^i_{k(t)} \\ |u_{k(t)}| &\leq u^k_{max},\ k(t) \in \mathscr{K} = \{1,\cdots,N\} \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the vector of process state variables and $u_k := [u^1_k \cdots u^m_k]^T$ denotes the vector of constrained manipulated inputs associated with the k-th control configuration. $k(t)$, which takes values in the finite index set, $\mathscr{K}$, represents a discrete state that indexes the vector fields $f_k(\cdot)$, $g^i_k(\cdot)$ as well as the manipulated inputs $u^i_k(\cdot)$. For each value that k assumes in $\mathscr{K}$, the process is controlled via a different set of manipulated inputs which define a given control configuration. Switching between the available N control configurations is controlled by a high-level supervisor that monitors the process and orchestrates, accordingly, the transition between the different control configurations in the event of control system failure. This in turn determines the temporal evolution of the discrete state, $k(t)$. The supervisor ensures that only one control configuration is active at any given time, and allows only a finite number of switches over any finite interval of time.

It is assumed that the origin is the equilibrium point of the nominal process and that the vector functions $f_k(\cdot)$ and $g^i_k(\cdot)$ are sufficiently smooth, for all k. The control objective is to stabilize the process of Eq.1 in the presence of actuator constraints and faults in the control system. The basic problem is how to coordinate switching between the different control configurations in a way that respects actuator constraints and guarantees closed-loop stability in the event of faults. Throughout the paper, the notation $|\cdot|$ is used to denote the Euclidean norm of a vector and the notation $L_f V$ denotes the Lie derivative of a scalar function, V, with respect to the vector field, f. To simplify the presentation of our results, we will focus only on the state feedback problem where measurements of all process states are available for all times.

2.2 *Motivating example*

In this section, we introduce a simple benchmark example that will be used throughout the paper to illustrate the design and implementation of the fault-tolerant control design methodology to be proposed in section 3. While the discussion will center around this example, we note that the proposed framework can be applied to more complex plants involving more complex arrangements of processing units. To this end, consider two well-mixed, non-isothermal continuous stirred tank reactors in series, where three parallel irreversible elementary exothermic reactions of the form $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} U$ and $A \xrightarrow{k_3} R$ take place, where A is the reactant species, B is the desired product and U, R are undesired byproducts. The feed to CSTR 1 consists of pure A at flow rate F_0, molar concentration C_{A0} and temperature T_0, and the feed to CSTR 2 consists of the output of CSTR 1 and an additional fresh stream feeding pure A at flow rate F_3, molar concentration C_{A03} and temperature T_{03}. Due to the non-isothermal nature of the reactions, a jacket is used to remove/provide heat to both reactors. Under standard modeling assumptions, a mathematical model of the plant can be derived from material and energy balances and takes the following form:

$$\begin{aligned}
\frac{dT_1}{dt} &= \frac{F_0}{V_1}(T_0 - T_1) + \sum_{i=1}^{3} R_i(C_{A1}, T_1) + \frac{Q_1}{\rho c_p V_1} \\
\frac{dC_{A1}}{dt} &= \frac{F_0}{V_1}(C_{A0} - C_{A1}) - \sum_{i=1}^{3} k_{i0} e^{\frac{-E_i}{RT_1}} C_{A1} \\
\frac{dT_2}{dt} &= \frac{F_1}{V_2}(T_1 - T_2) + \frac{F_3}{V_2}(T_{03} - T_2) \\
&\quad + \sum_{i=1}^{3} R_i(C_{A2}, T_2) + \frac{Q_2}{\rho c_p V_2} \\
\frac{dC_{A2}}{dt} &= \frac{F_1}{V_2}(C_{A1} - C_{A2}) + \frac{F_3}{V_2}(C_{A03} - C_{A2}) \\
&\quad - \sum_{i=1}^{3} k_{i0} e^{\frac{-E_i}{RT_2}} C_{A2}
\end{aligned} \tag{2}$$

where $R_i(C_{Aj}, T_j) = \frac{(-\Delta H_i)}{\rho c_p} k_{i0} \exp\left(\frac{-E_i}{RT_j}\right) C_{Aj}$, for $j = 1, 2$. T, C_A, Q, and V denote the temperature of the reactor, concentrations of the species A, rate of heat input/removal from reactor, and volume of reactor, respectively, with subscript 1 denoting CSTR 1 and subscript 2 denoting CSTR 2, ΔH_i, k_i, E_i, $i = 1, 2, 3$, denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively, c_p and ρ denote the heat capacity and density of the reactor. Using typical values of process parameters (values are omitted for brevity), CSTR 1, with $Q_1 = 0$, has three steady-states: two locally asymptotically stable and one unstable at $(T_{1s}, C_{A1s}) = (388.57\ K, 3.59\ kmol/m^3)$. The unstable steady-state of CSTR 1 corresponds to three steady-states for CSTR 2 (with $Q_2 = 0$), one of which is unstable at $(T_{2s}, C_{A2s}) = (429.24\ K, 2.55\ kmol/m^3)$.

The control objective is to stabilize both reactors at the (open-loop) unstable steady-states. To accomplish this objective under normal conditions (with no failures), we choose as manipulated inputs the rate of heat inputs, $u_1^1 = Q_1$, subject to the constraint $|Q_1| \leq u_{max}^{Q_1} = 2.7 \times 10^6\ KJ/hr$ and $u_1^2 = Q_2$, subject to the constraint $|Q_2| \leq u_{max}^{Q_2} = 2.8 \times 10^6\ KJ/hr$. As shown in Fig.1, each unit has a local control system with its sensors and actuators connected through a communication network. The local control systems in turn communicate with the plant supervisor (and with each other) through a plant-wide communication network. Note that in designing each control system, only measurements of the local process variables are used (for example, the controller for the second unit uses only measurements of T_2 and C_{A2}).

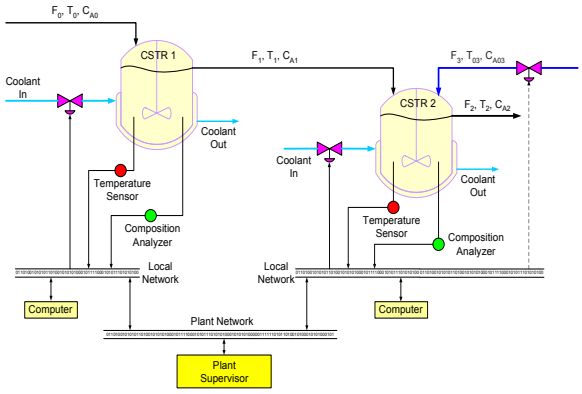

Fig. 1. Process flow diagram of two CSTRs in series.

The fault-tolerant control problem under consideration involves a catastrophic failure in both control systems after some time of startup, with the failure of the first unit being permanent. Our objective will be to preserve closed-loop stability of CSTR 2 by switching to an alternative control configuration involving, as manipulated variables, the rate of heat input, $u_2^1 = Q_2$, subject to the same constraint, and the inlet reactant concentration, $u_2^2 = C_{A03} - C_{A03s}$, subject to the constraint $|C_{A03} - C_{A03s}| \leq u_{max}^{C_{A03}} = 0.4\ kmol/m^3$ where $C_{A03s} = 3.0\ kmol/m^3$. The main question, which we address in the next section, is how to devise the switching and network communication logics in a way that ensures fault-tolerance in the second unit and, simultaneously, accounts for the inherent limitations in network resources and possible delays in fault-detection, communication and actuator activation.

3. FAULT-TOLERANT CONTROL SYSTEM DESIGN METHODOLOGY

In this section, we outline the main steps involved in the fault-tolerant control system design procedure, as applied to the reactors example of section 2.2. A major feature of the design methodology is the inherent coupling between the feedback, supervisory control and communication tasks.

(a) *Constrained feedback controller synthesis*

The main issue in this step is how to design the feedback control law for the fall-back configuration in CSTR 2 in a way that respects its actuators' constraints and guarantees closed-loop stability of this unit. The choice of the feedback law depends on our choice of

the communication policy. To explain this connection, we first note that a total failure in the control system of CSTR 1 will cause the states (C_{A1}, T_1) to move away from the desired steady-state for this unit. Therefore, unless the feedback controller for CSTR 2 is redesigned to account for this incoming "disturbance", the evolution of C_{A2}, T_2 will be affected causing them not to converge to the desired steady-state. To account for this disturbance, one option could be to send available measurements of C_{A1} and T_1, through the network, to the second unit and redesign its controller accordingly. From a communications cost point of view, this option may be costly since it requires continuous usage of the network after failure, which can adversely affect the performance of other units sharing the same communication medium due to bandwidth limitations and overall delays.

To reduce unnecessary network usage, we propose an alternative approach where we view the failure in CSTR 1 as a bounded non-vanishing disturbance affecting CSTR 2 and use the available process model of CSTR 1 to capture, or estimate, the size of this disturbance (by comparing, for example, the evolution of the process variables under the failed and well-functioning control configurations through simulations). In this formulation, in lieu of measurements of C_{A1} and T_1, only bounds on the disturbance size are needed and transmitted from CSTR 1 to CSTR 2, which involves using the network only at the failure time and not continuously. The disturbance information can then be used to design an appropriate robust controller to attenuate the effect of disturbances and enforce robust closed-loop stability in the second unit. To this end, we can rewrite the model of CSTR 2 in the following general form:

$$\dot{\bar{x}} = \bar{f}(\bar{x}) + \bar{G}(\bar{x})u + \bar{W}(\bar{x})\theta \qquad (3)$$

where $\bar{x} = [T_2 \;\; C_{A2}]^T$, $u = [Q_2 \;\; (C_{A03} - C_{A03s})]^T$ are the inputs of the fall-back configuration, and $\theta = [T_1 \;\; C_{A1}]^T$ are the time-varying, but bounded, disturbances. For this system, one possible choice for the controller is a variation of the class of bounded robust Lyapunov-based control laws proposed in (El-Farra and Christofides, 2003*a*) (see also (Lin and Sontag, 1991)) and has the general form:

$$u = -r(\bar{x}, u_{max}, \theta_b)(L_{\bar{G}}V)^T \qquad (4)$$

where $r(x, u_{max}, \theta_b) =$

$$\frac{L^*_{\bar{f}}V + \sqrt{(L^*_{\bar{f}}V)^2 + (u_{max}|(L_{\bar{G}}V)^T|)^4}}{|(L_{\bar{G}}V)^T|^2\left[1 + \sqrt{1 + (u_{max}|(L_{\bar{G}}V)^T|)^2}\right]} \qquad (5)$$

$L^*_{\bar{f}}V = L_{\bar{f}}V + \chi\theta_b|(L_{\bar{W}}V)^T|$, θ_b is the disturbance bound (maximum size of the norm of the disturbance vector), V is a control Lyapunov function, $L_{\bar{G}}V = [L_{g^1}V \;\; L_{g^2}V]$, $L_{\bar{W}}V = [L_{w^1}V \;\; L_{w^2}V]$ are row vectors, g^i, w^i are column vectors of the matrices $\bar{G}$ and $\bar{W}$, respectively, and $\chi > 0$ is a tuning parameter. The nonlinear gain function, $r(\cdot)$ in Eqs.4-5, which depends on the size of actuator constraints, the disturbance size and the type of configuration used, is shaped in a way that guarantees constraint satisfaction and robust closed-loop stability, with an arbitrary degree of attenuation of the effect of the disturbances, within a well-characterized region in the state space. The characterization of this region is given next.

(b) *Characterization of fault-recovery regions*

Having designed the feedback control law, the next step is to explicitly characterize the set of admissible states starting from where the constrained fall-back control configuration is stabilizing (fault-recovery region). As discussed in step (c), this characterization is necessary for the design of the switching policy that ensures fault-recovery. For the controller of Eqs.4-5, using a Lyapunov argument, one can show that the set

$$\Pi(u_{max}, \theta_b) = \{\bar{x} : L^*_{\bar{f}}V(\bar{x}) \leq u_{max}|(L_{\bar{G}}V)^T(\bar{x})|\} \qquad (6)$$

describes a region in the state space where the control action satisfies the constraints and the Lyapunov function decays monotonically along the trajectories of the closed-loop system (see (El-Farra and Christofides, 2003*a*) for the detailed mathematical analysis). Note that the size of this set depends both on the magnitude of the constraints and the size of the disturbance (which in turn depends on the failure time). For a given control configuration, one can use the above inequality to estimate the fault-recovery region by constructing, for example, the largest invariant subset of Π, which we denote by $\Omega(u_{max}, \theta_b)$.

(c) *Supervisory switching logic design*

In the general case where more than one fall-back control configuration is available, an important question is how to decide which of the available configurations can be activated following the failure of the primary configuration in order to preserve closed-loop stability. The key idea here is that the supervisor can only activate the configuration for which the closed-loop state, at the time of failure, is within the stability region. For the reactor example of section 2.2, and since a single fall-back configuration, (Q_2, C_{A03}), is considered, this implies that if failure occurs at any time T for which $\bar{x}(T) \in \Omega(u_{max}, \theta_b)$, then switching to this configuration guarantees preservation of closed-loop stability. The implementation of this switching rule requires monitoring the closed-loop state trajectory with respect to the fault-recovery region. The idea of constructing the switching logic based on the stability regions was first proposed in (El-Farra and Christofides, 2002) for the control of switched nonlinear systems. If failure occurs at times when the states are outside the stability region, our analysis suggests that either the constraints should be relaxed to enlarge the fault-recovery region of the given configuration, or additional fall-back control loops must be introduced. The latter option is ultimately limited by the maximum allowable number of control loops that can be designed for the process.

(d) *Design of the communication logic*

An essential element in the design of the fault-tolerant control system is the use of a communication medium that ensures fast and efficient transmission of information during failure events. Communication networks offer such a medium which is both fast (relative to the typically slow dynamics of chemical processes) and inexpensive (relative to current point-to-point connection schemes which require extensive cabling and higher maintenance time and costs). The ability of the network to fulfill this role, however, requires that the communication policy be devised in a way that respects the inherent limitations in network resources, such as bandwidth constraints and overall delays, by minimizing unnecessary usage of the network.

In step (a), we have already discussed how bandwidth constraints can be handled by formulating the problem as a robust control problem, where the failure in the first unit is treated as a bounded non-vanishing disturbance to the second unit. This policy avoids unnecessary overloading of the network while also guaranteeing fault-tolerance. The idea of using knowledge of the plant dynamics to manage the tradeoff between bandwidth limitations (which favor reduced communication of measurements) and optimum control performance (which favors increased communication of measurements) is conceptually aligned with the notion of minimum attention control (e.g., see (Brockett, 1997; Montestruque and Antsaklis, 2003)). In our work, however, this idea is utilized in the context of fault-tolerant control.

The second consideration in devising the communication logic is the issue of time-delays which typically result from the time sharing of the communication medium as well as the computing time required for the physical signal coding and communication processing. The characteristics of these time delays depend on the network protocols adopted as well as the hardware chosen. For our purposes here, we consider an overall fixed time-delay (which we denote by τ_{max}) that combines the contribution of several delays, including: (1) delays in fault-detection, (2) the time the local supervisor of unit 1 takes to compute the effective disturbance bounds (through simulations comparing the open-loop and closed-loop state evolution), (3) the time the local supervisor of unit 1 takes to send the information to the plant supervisor, (4) the time it takes the plant supervisor to forward the information to the local supervisor of unit 2, (5) the time it takes the local supervisor for unit 2 to compute the fault-recovery region for the given fall-back configurations using the information arriving from unit 1 and the time it takes for the supervisor's decision to reach and activate the appropriate fall-back configuration, and (6) the inherent actuator/sensor dead-times.

Failure to take such delays into account can result in activating the wrong control configuration and subsequent instability. For example, even though failure of a given loop may take place at $t = T$, the backup configuration will not be switched in before $t = T + \tau_{max}$. If the delay is significant, then the switching rule in part (c) should be modified such that the supervisor activates the configuration for which $\bar{x}(T + \tau_{max}) \in \Omega(u_{max}, \theta_b)$. This modification is yet another manifestation of the inherent coupling between the switching and communication logics. The implementation of the modified switching rule (which accounts for delays) requires that the local supervisor of unit 2 be able to predict where the trajectory will be at $t = T + \tau_{max}$ (e.g., through simulations using the process model) and check whether the state at this time is within the fault-recovery region of the fall-back configuration.

4. SIMULATION RESULTS

In this section, we revisit the example of section 2.2 to illustrate the implementation of the proposed fault-tolerant control methodology. To this end, the reactors are initialized at $(T_1(0), C_{A1}(0)) = (300\,K, 4.0\,mol/L)$, and $(T_2(0), C_{A2}(0)) = (440\,K, 4.0\,mol/L)$. Under normal operation (with no failures), each reactor is controlled by manipulating the rate of heat input using bounded nonlinear Lyapunov-based control algorithms (e.g., see (El-Farra and Christofides, 2002); details of the nominal controller design are omitted for brevity). It was verified that, in the absence of failures, the controllers stabilize both reactors at the desired steady-states. Consider now a catastrophic failure in both controllers (Q_1 and Q_2) at $t_{failure} = 5\,min$. In this case, both reactors revert to their open-loop modes and, consequently, the states are expected to move away from the desired steady-state as shown by the dashed lines in Figs.3(a)-(b) for CSTR 2.

Recall that the objective is to preserve closed-loop stability of CSTR 2 despite this failure. Using the proposed methodology, the supervisor of CSTR 1, at the failure time, runs both open- and closed-loop simulations of the process model to estimate the size of the disturbance that will affect CSTR 2 and transmits this information to CSTR 2 through the network. The maximum disturbance size is proportional to the largest discrepancy between the values of C_{A1}, T_1 in the failed and in the well-functioning modes. Using this information, the local supervisor designs a robust control law of the form of Eqs.4-5 to stabilize CSTR 2, using the fall-back configuration with (Q_2, C_{A03}) as the manipulated inputs, and constructs the associated fault-recovery region (the shaded area in Fig.2) with the aid of Eq.6. From Fig.2, we observe that failure occurs when the states are within the fault-recovery region. Therefore, assuming no delays in the fault-detection, computations and communication processing, when the fall-back controllers are activated, closed-loop stability is preserved and the states converge to the desired steady-state as shown by the solid lines in Figs.3(a)-(b).

When delay effects are taken into account, the dotted line in Fig.2 shows that if an overall delay of 3 min elapses between the failure and the activation of the

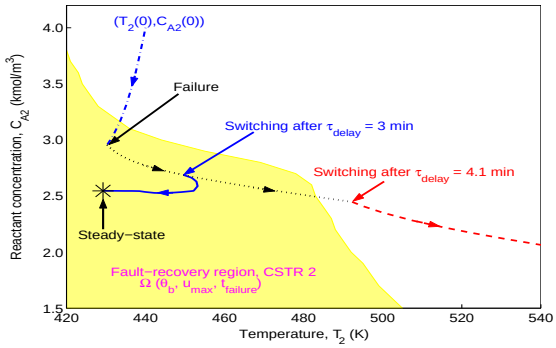

Fig. 2. Fault-recovery region of (Q_2, C_{A03}) config.

(Q_2, C_{A03}) configuration, the state at the end of the delay still resides within the fault-recovery region and, therefore, closed-loop stability can be preserved by switching as shown by the solid lines in Figs.3(c)-(d). By contrast, when the overall delay is 4.1 min, the state lies outside the fault-recovery region and the fall-back configuration cannot stabilize the system at the desired steady-state as can be seen from the dashed lines in Figs.3(c)-(d). Finally, examination of Fig.2 reveals how the interplay between the switching and communication tasks can be managed to ensure fault-tolerance. For example, the picture suggests that large communication delays can be tolerated by enlarging the fault-recovery region (e.g., by relaxing the constraints).

5. REFERENCES

Bao, J., W. Z. Zhang and P. L. Lee (2002). Passivity-based decentralized failure-tolerant control. *Ind. & Eng. Chem. Res.* **41**, 5702–5715.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics and constraints. *Automatica* **35**, 407–427.

Brockett, R. (1997). Minimum attention control. In: *Proceedings of 36th Conference on Decision and Control*. San Diego, CA. pp. 2628–2632.

El-Farra, N. H., A. Gani and P. D. Christofides (2004). Fault-tolerant control of process systems: Integrating supervisory and feedback control over networks. In: *Proceedings of 5th International Symposium on Advanced Control of Chemical Processes*. Hong Kong, P. R. China. pp. 784–789.

El-Farra, N. H. and P. D. Christofides (2001). Integrating robustness, optimality, and constraints in control of nonlinear processes. *Chem. Eng. Sci.* **56**, 1841–1868.

El-Farra, N. H. and P. D. Christofides (2002). Switching and feedback laws for control of constrained switched nonlinear systems. In: *Lecture Notes in Computer Science Series*. Vol. 2289. Tomlin, C. J. and M. R. Greenstreet (Eds.), Berlin, Germany: Springer-Verlag. pp. 164–178.

El-Farra, N. H. and P. D. Christofides (2003*a*). Bounded robust control of constrained multivariable nonlinear processes. *Chem. Eng. Sci.* **58**, 3025–3047.

El-Farra, N. H. and P. D. Christofides (2003*b*). Coordinating feedback and switching for control of

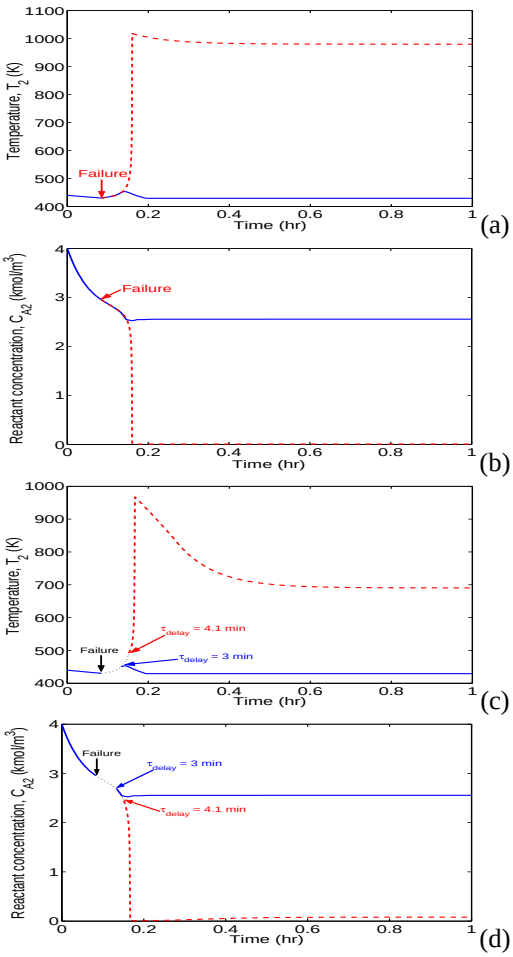

Fig. 3. Evolution of closed-loop state profiles when the fall-back controllers are not activated after failure (a–b, dashed), activated with no delays (a–b, solid), activated after total delays of 3 min (c–d, solid) and 4.1 min (c–d, dashed).

hybrid nonlinear processes. *AIChE J.* **49**, 2079–2098.

Lin, Y. and E. D. Sontag (1991). A universal formula for stabilization with bounded controls. *Systems & Control Letters* **16**, 393–397.

Montestruque, L. A. and P. J. Antsaklis (2003). On the model-based control of networked systems. *Automatica* **39**, 1837–1843.

Patankar, R. (2003). A model for fault-tolerant networked control system using TTP/C communication. In: *Proceedings of American Control Conference*. Denver, CO. pp. 533–537.

Tipsuwan, Y. and M.-Y. Chow (2003). Control methodologies in networked control systems. *Contr. Eng. Prac.* **11**, 1099–1111.

Walsh, G. C., H. Ye and L. G. Bushnell (2002). Stability analysis of networked control systems. *IEEE Trans. Contr. Syst. Tech.* **10**, 438–446.

Willsky, A. S. (1998). A survey of design methods for failure detection in dynamic systems. *Automatica* **12**, 601–611.

Yang, G. H., S. Y. Zhang, J. Lam and J. Wang (1998). Reliable control using redundant controllers. *IEEE Trans. Automat. Contr.* **43**, 1588–1593.