Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Advanced design scheme for fault tolerant distributed networked control systems [★]

S. X. Ding [*] P. Zhang [*] Ch. Chihaia [*] W. Li [*] Y. Wang [*]
E. L. Ding [**]

[*] *Institute for Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, 47057 Duisburg, Germany*
[**] *Department of Physical Engineering, University of Applied Sciences Gelsenkirchen, 45877 Gelsenkirchen, Germany*

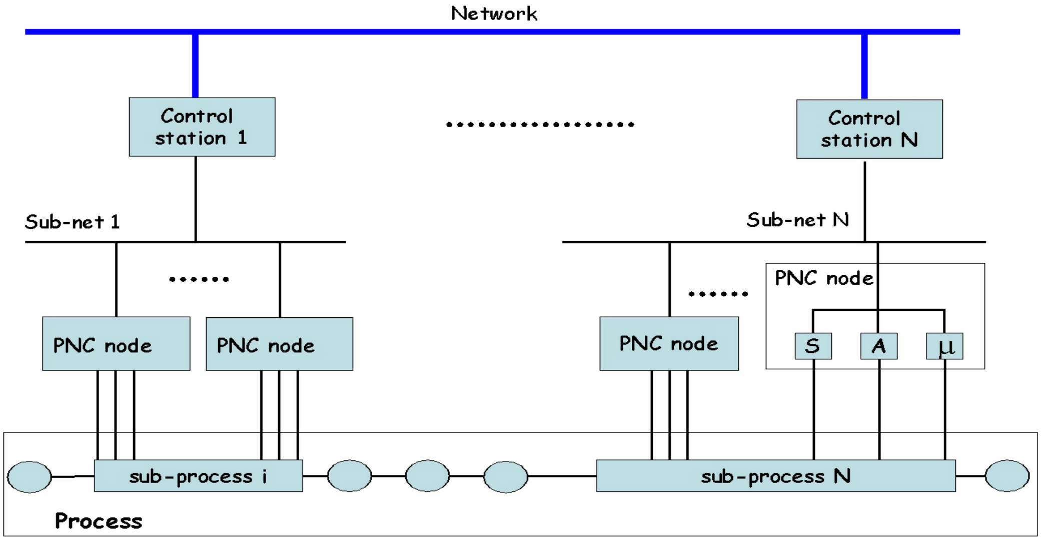
Abstract: This paper addresses the integrated design of fault tolerant distributed networked control systems (NCS). The NCS under consideration consists of two levels. At the lower level, sensors, actuators and local controllers are embedded and networked by sub-nets. They coordinated and supevised by the control stations located at the higher level. The core of the design scheme is the integrated design of communication, control and fault diagnosis systems in a multilayer structure.

Keywords: Fault tolerant systems; networked control systems; observer based fault tolerant scheme; fault diagnosis; periodic systems.

1. INTRODUCTION

The wide application of networked control systems (NCS) marks the state of the art in the area of automatic control. In the past decade, rapid development of microelectronic, information and communication technologies enhanced networking of intelligent sensors, actuators, controllers and microprocessors and accelerated the application of NCSs in major industrial sectors. This trend is strongly driven by the industrial needs for highly distributed automatic systems and networked embedded systems Furrer [2003], Moyne and Tilbury [2007].

Integrating networks into automatic control systems can significantly increase the automation degree to meet the demands for high productivity and product quality, and allows a flexible system configuration with less wiring and an easy maintenance. Many different types of networks have been promoted for different applications, for instance, CAN, Ethernet, WLAN, etc. Remarkably different from classical control systems, the performance and behavior of the NCSs considerably depend on the technical characteristics of the network. In addition, accompanied with the growth of the integration and automation degree the overall failure rate will significantly increase.



PNC: Process Near Components including Sensors, Actuators and Microprocessor

Fig.1: Schematic description of a distributed NCS

[★] This work was supported in part by the EU grant IST-004303 NeCST.

A most critical and important issue surrounding the design of distributed NCSs with the successively increasing complexity is to meet the requirements on system reliability and dependability, while guaranteeing a high system performance over a wide operating range Patton et al. [2007]. In this paper, we shall briefly report our efforts in developing advanced design schemes for fault tolerant NCSs with the structure as shown in Fig.1. Our work is a part of the European project entitled *N*e*t*worked *C*ontrol *S*ystems *T*olerant to Faults (NeCST), whose objective is to design the NCSs that are tolerant to possible process, component and network faults.

Application of the NCS sketched in Fig.1 to automatic control of distributed processes can be often observed in many industrial sectors like process industry, manufacturing, transport and traffic systems etc. It is distributed, hierarchically constructed and consists of (a) a great number of PNC nodes, into which sensors, actuators and microprocessors are integrated (b) N control stations (CS), each of which coordinates and supervises a set of PNC nodes (c) a communication system that networks the CSs at a higher level and the PNC nodes at the lower level with the corresponding CS.

Recently, research on NCS receives considerably enhanced attention in the automatic control community. The major focuses of the research activities are on system performance analysis and controller design regarding to the technical properties of the network, which are expressed in terms of the so-called QoS (Quality of Service) parameters of the network. The major QoS parameters are data transmission delays, jitter, packet loss rate and network failure rate. Significant results have been published, see for instance Elia and Mitter [2001], Zhang et al. [February 2001], Lian et al. [2001], Ishii and Francis [2002], Tipsuwan and Chow [2003], Montestruque and Antsaklis [2004].

Studies in the past have revealed that for a given network the data transmission delays, jitter and packet loss rate strongly depend on the network load Lian et al. [2001], Furrer [2003]. In particular, for those networks like Ethernet or WLAN, the QoS parameters may change rapidly as the network load increases. In a typical distributed industrial NCS, the number of the nodes and the networked sensors, actuators as well as microprocessors is, different from the open structured NCS like an Internet based NCS, constant during the normal process operation and only varies in case of faults. On the other hand, to meet high control performance and reliability requirements, the QoS parameters should satisfy the requirements of the highest CoS (Class of Service). The major objective of our study is to integrate the design of the fault tolerant control and communication systems by means of a trade-off between the Quality of (system) Performance (QoP) and the QoS.

2. OUTLINE OF THE FAULT TOLERANT NCS DESIGN SCHEME

In this section, we shall highlight the system structure from the viewpoints of fault tolerant control (FTC), fault detection and isolation (FDI) and communication, outline the basic ideas behind the fault tolerant NCS design scheme and finally formulate the problems to be addressed.

2.1 Structure of fault tolerant control scheme

To achieve high reliability and to meet the demanded control performance, the fault tolerant scheme sketched in Fig.2 is proposed, which consists of three functional layers.

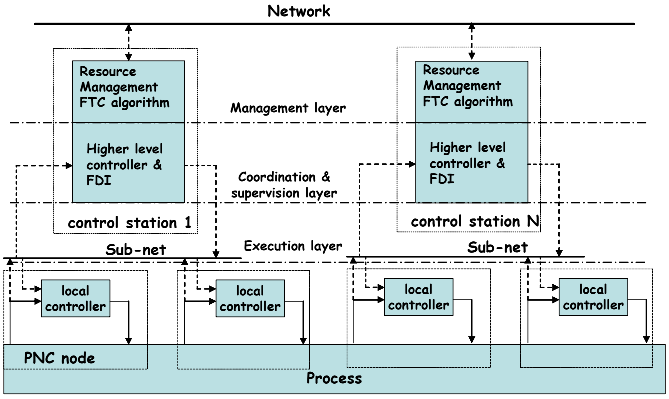


Fig.2: fault tolerant control structure

Execution layer: at this layer, PNC nodes are integrated with embedded local controllers. The local controllers serve for three purposes: (a) they should ensure the overall system stability in the totally decentralized mode, i.e. in case that the communication between the control stations and the PNC nodes is broken down (b) their implementation should simplify the design and implementation of the higher level controllers, since the process together with its local controllers can be considered as a stable plant (c) they relieve the communication between the execution and the coordination/supervision layers with loss of real time performance. Simple FDI units will also be integrated into this layer, which allow an early detection of large sized faults in the local sensors, actuators and process components.

Coordination & supervision layer: embedded in the CSs, advanced control schemes and comprehensive FDI algorithms are implemented in the higher level controllers and FDI units. The core of the higher level controllers and FDI units is a distributed observer bank that is driven by the sensor signals received from the PNC nodes and delivers an estimation of the process state variables. The control commands for the local controllers and the residual signals for the FDI purpose are generated based on the state estimation.

Management layer: in our study, FTC is implemented in the context of resource management Paoli [2004]. Any component, sensor or actuator or process component, is defined as system resource that is needed for some functionality. A fault in one component will be considered as a loss of the corresponding resource or redundancy and activate a resource re-allocation, making use of the available redundancy, to ensure the system operation. The resource management scheme and the associated FTC algorithms will activate, in case of a fault, a re-configuration of the controllers, FDI units and the communication protocols.

2.2 Multi-layer communication structure

A key issue surrounding the design and implementation of the above-described fault tolerant system is to guarantee the required system QoP by providing the needed CoS of the networks. In NCSs, the data transmission is often regulated based on the ISO/OSI three-layer model, including (a) physical layer (b) data link layer and (c) application layer. Fig.3 shows the basic principle of the three-layer model based data transmission. The physical layer is standardized regarding to the hardware and operating system. Hence, at this layer no design freedom is available for the designer. Differently, at the data link layer, also called medium access control (MAC), or at the application layer, the designer is able to implement a scheduler to guarantee, on the one side, the required real-time performance and regulate, on the other side, the QoS parameters of the network.

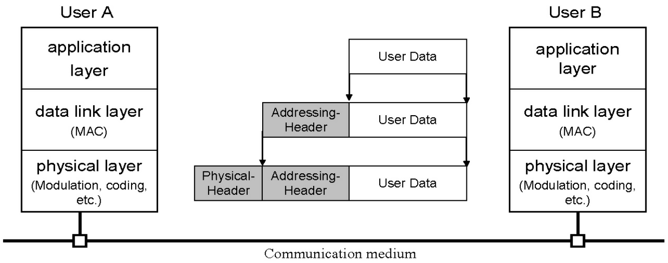


Fig.3: Schematic description of a data transmission model

For the design and implementation of the fault tolerant NCS, we propose to structure the application and MAC layers into three sub-layers, corresponding to the three functional layers sketched in Fig.2. From the communication viewpoint, the data exchanges between two users (nodes) at each sub-layer are as follows:

At the execution layer: the communication between the CS and the associated PNC nodes is executed via sub-net. To ensure the required real time behavior and reliability, the communication at this layer will be regulated by a scheduler, which will be integrated either into the MAC or into the application layer. The communication between a CS and the associated PNC nodes will operate in a master-slave mode with the CS as the master. There exists no communication between the PNC nodes.

At the coordination and supervision layer: the communication between the CSs serves as synchronization and execution of the control, monitoring and communication actions. The data exchanges at this layer are periodic and regulated by a protocol in the token passing manner.

At the management layer: the data exchange at this layer will be activated if a fault is identified and a resource re-allocation becomes needed, i.e. it is event-driven. It serves as a distributed computation of the resource re-allocation algorithms.

2.3 Basic idea and problem formulation

The basic idea of the fault tolerant NCS design scheme is the integrated design of the multilayer fault tolerant control and communication systems. Corresponding to the different control and FDI functional layers with different requirements on the real time behavior and on the way of data exchanges, different scheduling strategies will be used for the data transmission. As sketched in Fig.4, the core of this scheme is a distributed observer bank and a resource monitor bank. The former provides the controllers (both the local and higher level controllers) and FDI units the needed information for control and FDI actions. In case of faults, it will also provide the management layer with the knowledge of the faults.

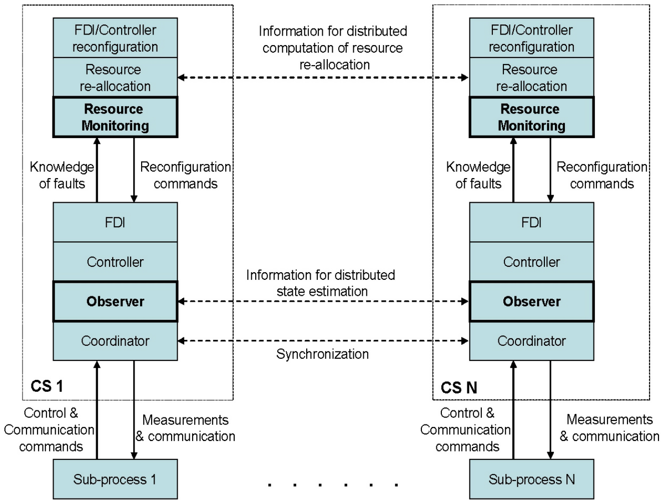


Fig.4: Schematic description of the design scheme

Due to the limited space, in the next sections we shall pay major attention to the construction of the fault tolerant (FT) NCS with a focus on the distributed observer bank. The methods used for the system design will only be briefly described.

3. CONSTRUCTION OF THE FT NCS

3.1 Process model

Suppose that the process under consideration consists of N sub-processes modelled by the discrete time system

$$x_i(k_o+1) = A_{ii}x_i(k_o) + B_{ii}u_i(k_o) + E_{d,i}d(k_o) \qquad (1)$$

$$+\sum_{j\neq i}^{N} A_{ij}x_j(k_o), i = 1, \cdots, N$$

$$x_i \in R^{n_i}, u_i \in R^{q_i}, n = \sum_{i=1}^{N} n_i, q = \sum_{i=1}^{N} q_i$$

with the sampling time T_o that is sufficiently small so that (1) well describes the continuous time process. In (1), x_i, u_i stand for the state and input vectors of the i-th sub-process, $d \in R^{k_d}$ for the unknown input vector, $A_{ij}, B_{ii}, E_{d,i}$ are known matrices of appropriate dimensions. Suppose that the sensors of the i-th subsystem are modelled by

$$y_i(k_o) = C_i x_i(k_o) + D_i u_i(k_o) + F_i d(k_o) \qquad (2)$$

where $y_i \in R^{m_i}$, C_i, D_i, F_i are known matrices. Below, we shall use the notation "subsystem" to represent the composite of a sub-process, the associated CS and the corresponding PNC nodes.

3.2 Communication scheduling strategy

As mentioned in the last section, the communication between a CS and the associated PNC nodes and the communication among the CSs will be regulated in different ways.

Roughly speaking, the objective of designing a scheduler for the regulation of the communication between a CS and the associated PNC nodes is to (a) ensure the deterministic data transmission behavior (b) guarantee the required QoS values. For our purpose, we shall apply the static cyclic schedule, which can be dynamically (online) re-constructed in case that faults are identified and the resource management activates a re-configuration of the control, FDI and communication structures and algorithms. The basic idea behind the fault tolerant scheduler is the individual reservation of the channel capacity for the three major actions: (a) transmission of the sensor signals from the PNC nodes to the CS (b) transmission of control commands from the CS to the PNC nodes (c) implementation of standard communication strategies to fulfill the requirements on the (low) packet loss rate, high reliability and to ensure the system synchronization. In our study, the scheduler is designed based on the time-division multiple-access (TDMA) strategy.

Let $\tau_{i,\max}$ be the maximum data transmission time (including physical transmission and software operation times) between any two nodes within the i-th sub-system. Define a time slot $\geq \tau_{i,\max}$. The data transmission between the CS and the PNC nodes will be periodic. In one cycle, the following time slots are reserved for (a) transmission of sensor data with m_i time slots (b) transmission of control commands with q_i time slots (c) implementation of the communication strategy with $h_{i,c}$ time slots. $h_{i,c}\,(\geq 1)$ is an integer and $h_{i,c}\tau_{i,\max}$ is reserved for those actions like special coding schemes, acknowledgement of receiving data, asking for repeating sending, sending synchronization signals etc. Let $T_{i,c}$ be the cyclic time, which is set to be $T_{i,c} \geq (m_i + q_i + h_{i,c})\,\tau_{i,\max}$. The above-mentioned actions are coordinated by the CS in the role of a master.

Using a communication protocol, the data exchanges among the CSs will be coordinated in the token passing manner. Assume that the i-th CS receives the sensor data at time instant t. It will update the state estimation and activate the further data exchanges: (a) The i-th CS transmits the updated estimate and the associated data to the rest CSs (b) those CSs update their observers and (c) transmit the update results to the other CSs. The time instants, at which the CSs receive their sensor data, will be scheduled by the protocol to avoid collision. Without loss of generality, the communication will be synchronized and regulated to be

$$l_1 T_{1,c} = l_2 T_{2,c} = \cdots = l_N T_{N,c} = T$$

with integer $l_i, i = 1, \cdots, N$. To illustrate the scheduling strategies schematically, in Fig.5 a simple example is sketched.

3.3 Execution layer

At the execution layer, local feedback control loops are integrated, equipped with sensors and actuators. One of the basic functions at this layer is the execution of control commands. **The local control law** is set to be

$$u_i(z) = K_i(z) y_i(z) + u_{i,com} \tag{3}$$

where $K_i(z)$ stands for some simple structured controller like P or PI controller, $u_{i,com}$ represents the control command sent by the i-th CS, which will be described in more detail in the subsequent subsections. Note that $u_{i,com}$ is constant during one cycle. For the sake of simple notation, we denote the closed-loop model of the i-th sub-system with its local controller (3) by

$$\bar{x}_i(k_o+1) = \bar{A}_{ii}\bar{x}_i(k_o) + \sum_{j \neq i}^{N} \bar{A}_{ij}\bar{x}_j(k_o) \tag{4}$$

$$+\bar{B}_{ii}u_{i,com} + \bar{E}_{d,i}d(k_o)$$

$$y_i(k_o) = \bar{C}_i\bar{x}_i(k_o) + \bar{D}_i u_{i,com}(k_o) + \bar{F}_i d(k_o) \tag{5}$$

where $\bar{x}_i(k_o)$ denotes the composite of the state variables of the i-th sub-process and the local controller $K_i(z)$ and $\bar{A}_{ij}, \bar{B}_{ii}, \bar{E}_{d,i}, \bar{C}_i, \bar{D}_i,\ \bar{F}_i$ the corresponding system matrices. We denote the overall process model (with local controllers) by

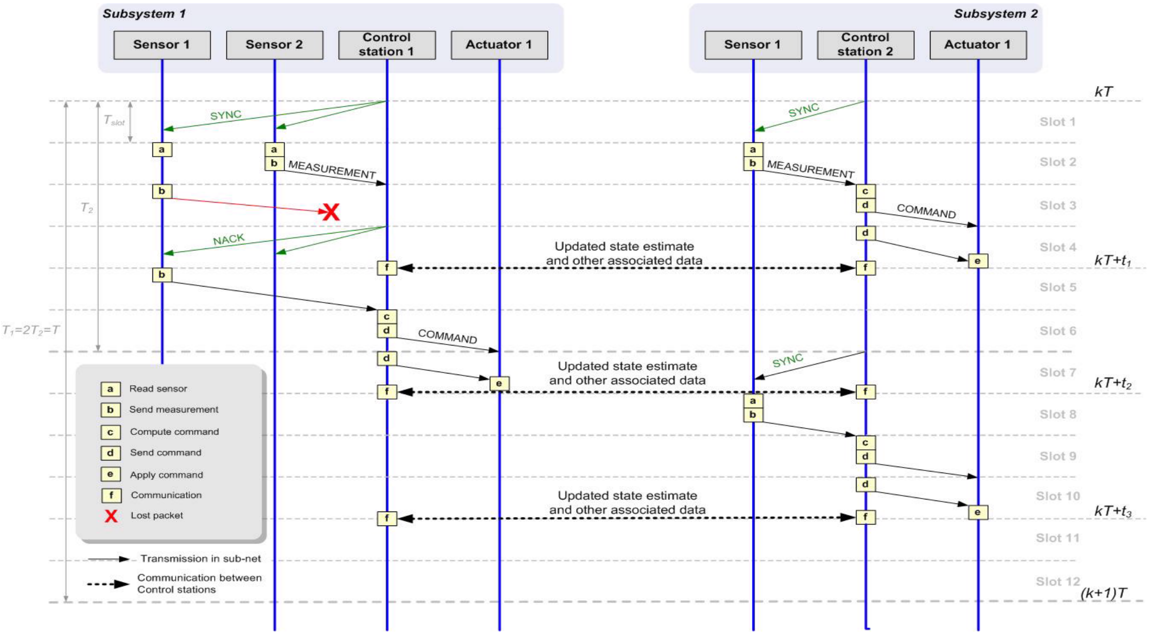


Fig.5: An example to illustrate the scheduling strategy

$$\bar{x}(k_o+1) = \bar{A}\bar{x}(k_o) + \bar{B}u_{com}(k_o) + \bar{E}_d d(k_o) \in \mathcal{R}^{n\times n} \quad (6)$$

$$\bar{A} = \left[\bar{A}_{ij}\right]_{n\times n}, \bar{B} = \left[\bar{B}_{ij}\right]_{n\times q}, \bar{E}_d = \left[\bar{E}_{d,i}\right]_{n\times k_d} \quad (7)$$

A PNC node will also receive additional data, $J_{th,j_i}, j_i = 1, \cdots, m_i$, as thresholds for the early detection of large sized faults, i.e.

$$|y_{j_i}| > J_{th,j_i} \Longrightarrow \text{a (large sized) fault}$$

where y_{j_i} denotes the j_i-th sensor signal.

3.4 Coordination and supervision layer

To begin with, we first describe the system model used for constructing the distributed observer bank. We denote the time instants in $[kT, (k+1)T)$, at which the local controllers at the execution layer receive the control commands from the CSs, by $kT + t_j, j = 1, \cdots, \sum_{p=1}^{N} l_p q_p$. Then the i-th sub-system can be written as

$$\bar{x}_i(kT+t_{j+1}) = \tilde{A}_{ii}(j)\bar{x}_i(kT+t_j) + \tilde{B}_{ii}(j)u_i(kT+t_j)$$

$$+\tilde{E}_{d,i}(j)\bar{d}(kT+t_j) + \sum_{q\neq i}^{N} \begin{pmatrix} \tilde{A}_{iq}(j)\bar{x}_q(kT+t_j) \\ +\tilde{B}_{iq}(j)u_q(kT+t_j) \end{pmatrix} \quad (8)$$

Note that the time between two time instants, say t_j, t_{j+1}, may be varying. We denote it by

$$t_{j+1} - t_j = \alpha_j T_o$$

As a result, the system matrices in (8) satisfy

$$\tilde{A}_{iq}(j) = \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{\alpha_j - 2} \begin{bmatrix} \bar{A}_{1q} \\ \vdots \\ \bar{A}_{Nq} \end{bmatrix}, \alpha_j > 1$$

$$\tilde{B}_{iq}(j) = \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{\alpha_j - 2} \begin{bmatrix} \bar{B}_{1q} \\ \vdots \\ \bar{B}_{Nq} \end{bmatrix}, \alpha_j > 1$$

$$\tilde{E}_{d,i}\bar{d}(kT+t_j) = \sum_{p=2}^{\alpha_j} \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{p-2}\bar{E}_d\bar{d}(kT+pT_o)$$

$$\tilde{A}_{iq}(j) = \bar{A}_{iq}, \tilde{B}_{iq}(j) = \bar{B}_{iq}, \tilde{E}_{d,i}(j) = \bar{E}_{d,i} \text{ for } \alpha_j = 1$$

with $i, q = 1, \cdots, N$. Suppose that in the time interval $[kT+t_j, kT+t_{j+1})$ the i-th CS receives measurement data, denoted by $v_i(kT+t_{j+1})$, from the local sensors and will transmit a control command $u_i(kT+t_{j+1})$ to the local actuators. Taking into account the possible delay due to the execution of the defined communication actions, the output model is described by

$$v_i(kT+t_{j+1}) = y_i(kT+t^i_{s,p}) = C_i x_i(kT+t^i_{s,p}) \quad (9)$$

with $kT + t^i_{s,p}$ denoting the time instant, at which the local sensors send their measurement to the i-th CS. The subscripts s, p stand for sensor and the sequence number of the (sensor) data transmission to the i-th CS during the time interval $[kT, (k+1)T)$. Depending on the communication actions and coordination between the subsystems, there are two possible cases: (I) $t^i_{s,p} - t_j = \beta^i_{j,p}T_o$ (II) $t_{j-1} \leq t^i_{s,p} < t_j, t^i_{s,p} - t_{j-1} = \beta^i_{j-1,p}T_o$. In Case I,

$$v_i(kT+t_{j+1}) = \tilde{C}_{ii}(j)\bar{x}_i(kT+t_j) + \tilde{D}_{ii}(j)u_i(kT+t_j)$$

$$+\sum_{q\neq i}^{N} \begin{pmatrix} \tilde{C}_{iq}(j)\bar{x}_q(kT+t_j)+ \\ \tilde{D}_{iq}(j)u_q(kT+t_j) \end{pmatrix} + \tilde{F}_{d,i}(j)\bar{d}(kT+t_j) \quad (10)$$

$$\tilde{C}_{iq}(j) = \bar{C}_i \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{\beta^i_{j,p}-2} \begin{bmatrix} \bar{A}_{1q} \\ \vdots \\ \bar{A}_{Nq} \end{bmatrix}, \beta^i_{j,p} > 1$$

$$\tilde{D}_{iq}(j) = \bar{C}_i \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{\beta^i_{j,p}-2} \begin{bmatrix} \bar{B}_{1q} \\ \vdots \\ \bar{B}_{Nj} \end{bmatrix} + \bar{D}_{ij}$$

$$\bar{D}_{iq} = \bar{D}_i \text{ for } i = q \text{ and } \bar{D}_{iq} = 0 \text{ for } i \neq q, \beta^i_{j,p} > 1$$

$$\tilde{F}_{d,i}\bar{d}(kT+t_j) = \sum_{p=2}^{\beta^i_{j,p}} \left[\begin{array}{ccc} \bar{A}_{i1} & \cdots & \bar{A}_{iN} \end{array}\right] \bar{A}^{p-2}\bar{E}_d\bar{d}(kT+pT_o)$$

$$\tilde{C}_{iq}(j) = \bar{C}_i\bar{A}_{iq}, \tilde{D}_{ii}(j) = \bar{C}_i\bar{B}_{ii} + \bar{D}_i, \tilde{D}_{iq}(j) = 0, i \neq q$$

$$\tilde{F}_{d,i}(j) = \bar{C}\bar{E}_{d,i} + \bar{F}_{d,i} \text{ for } \beta^i_{j,p} = 1$$

with $i, q = 1, \cdots, N$. In Case II, $v_i(kT+t_{j+1})$ is given by

$$\tilde{C}_{ii}(j)\bar{x}_i(kT+t_{j-1}) + \tilde{D}_{ii}(j)u_i(kT+t_{j-1})+$$

$$\sum_{q\neq i}^{N} \begin{pmatrix} \tilde{C}_{iq}(j)\hat{x}_q(kT+t_{j-1})+ \\ \tilde{D}_{iq}(j)u_q(kT+t_{j-1}) \end{pmatrix} + \tilde{F}_{d,i}(j)\bar{d}(kT+t_{j-1}) \quad (11)$$

Based on (8), (10) or (11) and on the assumption that $\hat{\bar{x}}_q(kT+t_j), q = 1, \cdots, N$ are available, the construction and execution of the distributed observer bank can be realized as follows:

Computation of the estimate $\hat{\bar{x}}_i(kT+t_{j+1})$:

$$\hat{\bar{x}}_i(kT+t_{j+1}) = \tilde{A}_{ii}(j)\hat{\bar{x}}_i(kT+t_j) + \tilde{B}_{ii}(j)u_i(kT+t_j)$$

$$+\sum_{q\neq i}^{N} \left(\tilde{A}_{iq}(j)\hat{\bar{x}}_q(kT+t_j) + \tilde{B}_{iq}(j)u_q(kT+t_j)\right) +$$

$$L_i(j)(v_i(kT+t_{j+1}) - \hat{v}_i(kT+t_{j+1})) \quad (12)$$

where $r_i(kT+t_{j+1}) = v_i(kT+t_{j+1}) - \hat{v}_i(kT+t_{j+1})$ builds the so-called residual signal that will be used for the FDI purpose and $L_i(j)$ is the observer gain. For Case I

$$\hat{v}_i(kT+t_{j+1}) = \tilde{C}_{ii}(j)\hat{\bar{x}}_i(kT+t_j) + \tilde{D}_{ii}u_i(kT+t_j)$$

$$+\sum_{q\neq i}^{N} \left(\tilde{C}_{iq}(j)\hat{\bar{x}}_q(kT+t_j) + \tilde{D}_{iq}(j)u_q(kT+t_j)\right)$$

and for Case II

$$\hat{v}_i(kT+t_{j+1}) = \tilde{C}_{ii}(j)\hat{\bar{x}}_i(kT+t_{j-1}) + \tilde{D}_{ii}u_i(kT+t_{j-1})$$

$$+\sum_{q\neq i}^{N} \left(\tilde{C}_{iq}(j)\hat{\bar{x}}_q(kT+t_{j-1}) + \tilde{D}_{iq}(j)u_q(kT+t_{j-1})\right)$$

Computation of the control command $u_i(kT+t_{j+1})$

$$u_i(kT+t_{j+1}) = K_i(j)\hat{\bar{x}}_i(kT+t_{j+1}) + w_{i,ref}$$

with $w_{i,ref}$ as a reference signal and **threshold** $J_{th,j_i}, j_i = 1, \cdots, m_i$.

Data transmission: the i-th CS sends $r_i(kT+t_{j+1})$, $u_i(kT+t_{j+1})$ and $\hat{\bar{x}}_i(kT+t_{j+1})$ to the q-th CS, $q = 1, \cdots, N, q \neq i$.

Computation of the estimate $\hat{\bar{x}}_q(kT+t_{j+1})$ in the q-th CS:

$$\begin{aligned}\hat{\hat{x}}_q(kT+t_{j+1}) &= \tilde{A}_{qq}(j)\hat{\hat{x}}_q(kT+t_j) + \tilde{B}_{qq}(j)u_i(kT+t_j)\\ &+\sum_{p\neq q}^{N}\left(\tilde{A}_{qp}(j)\hat{\hat{x}}_p(kT+t_j)+\tilde{B}_{qp}(j)u_p(kT+t_j)\right)\\ &+L_q(j)r_i(kT+t_{j+1}) \qquad (13)\end{aligned}$$

Data transmission: the q-th CS, $q = 1, \cdots, N$, sends $\hat{\hat{x}}_q(kT+t_{j+1})$ to the p-th CS, $p = 1, \cdots, N, p \neq q$. As a result, $\hat{\hat{x}}_i(kT+t_{j+1}), i = 1, \cdots, N$, are available at each CS for the next update.

Further actions at this layer include **data transmission**: the i-th CS sends $u_i(kT+t_{j+1}), J_{th,j_i}$, in packet, to the associated PNC nodes, **implementation of the observer based FDI scheme in each CS, handling of missing packets.** Note that if the transmission time of a data packet from a PNC node to the associated CS, say the i-th CS, is larger than $T_{i,c}$, the packet will be treated as missing.

3.5 Management layer

Resource monitors are driven by the knowledge of the faults provided by the FDI units (Fig.4). It is realized in form of a database, in which the available sensors (including observers as soft sensors), actuators, communication systems, process components together with their redundancy are clustered in terms of their role for executing a defined functionality (control, FDI, etc.). Resource management and re-allocation will be formulated as an optimization problem and solved by means of an optimization algorithm Paoli [2004].

4. DESIGN METHODS AND ASSOCIATED TOOLS

To realize the scheduling and synchronization strategies, the schemes proposed by Walsh and Hong [2001] and Johannessen [2004] can be used. For the design of the local controllers, the decentralized control schemes described in Bernussou and Titli [1982] are available. It is evident that (8), (10) or (11) describe a periodic system with period T. The key to the design of the fault tolerant NCS is the design of (periodic) observers given by (12) and (13). For this purpose, we can use, for instance, the methods proposed by Bittanti and Colaneri [1996], Bittanti and Cuzzola [2001]. As for periodic FDI, controller design and handling of missing packets, we refer the reader to Zhang et al. [2005], Zhang and Ding [2007], Bittanti and Colaneri [2000] and Zhang et al. [2004]. In Blanke et al. [2003] and Paoli [2004], advanced FTC methods are given, which are useful for the design of FTC units.

5. CONCLUSION

In this paper, we have proposed a design scheme for the fault tolerant distributed NCS. The core of this scheme is the integrated design of communication, control and fault diagnosis systems in a multilayer structure.

REFERENCES

J. Bernussou and A. Titli. *Interconnected Dynamic Systems: Stability, Decomposition and Decentralisation.* North Holland, 1982.

S. Bittanti and P. Colaneri. Periodic control. *in John Wiley Encyclopaedia on Electrial and Electronic Engineering*, 16:2–16, 2000.

S. Bittanti and P. Colaneri. Analysis of discrete-time linear periodic systems. *Control and Dynamics Systems*, 78: 313–339, 1996.

S. Bittanti and F. A. Cuzzola. An LMI approach to periodic discrete-time unbiased filtering. *Systems and Control Letters*, 42:21–35, 2001.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control.* Springer, 2003.

N. Elia and S.K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, 2001.

F. J. Furrer. *Industrieautomation mit Ethernet-TCP/IP und Web-Technologie.* Hüthig Verlag, 2003.

H. Ishii and B.A. Francis. *Limited Data Rate in Control Systems with Networks.* Springer, Berlin, 2002.

S. Johannessen. Time synchronization in a local area network. *IEEE Control Systems Magazine*, pages 61–69, 2004.

F.L. Lian, J.R. Moyne, and D.M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet and DeviceNet. *IEEE Control Systems Magazine*, pages 66–83, 2001.

L. A. Montestruque and P. Antsaklis. Stability of model-based networked control systems with time-varying transmission times. *IEEE Transactions on Automatic Control*, 49(9):1562–1572, 2004.

J. R. Moyne and D. M Tilbury. The emergence of industrial control networks for matufacturing control, diagnostics, and safety data. *Proc. of the IEEE*, 95:29–47, 2007.

A. Paoli. *Fault Detection and Fault Tolerant Control for Distributed Systems: A General Framework.* PhD thesis, University of Bologna, 2004.

R. J. Patton, Kamphampati C, Casavola A, Zhang P, Ding S, and Sauter D. A generic strategy for fault-tolerance in control systems distributed over a network. *European J. Control*, 13:280–296, 2007.

Y. Tipsuwan and M.Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11:1099–1111, 2003.

G.C. Walsh and Y. Hong. Scheduling of networked control systems. *IEEE Control Systems Magazine*, pages 57–65, 2001.

P. Zhang and S. Ding. Disturbance decoupling in fault detection of linear periodic systems. *Automatica*, 43: 1410–1417, 2007.

P. Zhang, S.X. Ding, P.M. Frank, and M. Sader. Fault detection of networked control systems with missing measurements. In *Proceedings of the Asian Control Conference*, pages 1257–1262, Melbourne, Australien, 2004.

P. Zhang, S.X. Ding, G.Z. Wang, and D.H. Zhou. Fault detection of linear discrete-time periodic systems. *IEEE Transactions on Automatic Control*, 50(2):239–244, 2005.

W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, pages 84–99, February 2001.