DESIGN METHOD FOR THE CONTROL OF SYSTEMS IN FAULT CONDITIONS

Eugen Iancu, Matei Vînătoru

Automation Departmen, University of Craiova
ROMANIA
Eugen.Iancu@automation.ucv.ro, vinatoru@automation.ucv.ro

Abstract: *In this paper, the authors propose a practical design method useful in control and supervising of control system in presence of process faults. The fault detection and isolation (FDI) problem is an inherently complex one. Because of this reason, we have considered the case when one or more parameters of process are modified. The immediate goals is to preserve the stability of process and, if is possible, to control the process in a slightly degraded manner.*

Keywords: *fault diagnosis, fault detection, fault isolation, reconfiguration, control system.*

1. INTRODUCTION

Fault detection and isolation (FDI) techniques have been widely used in process industry to detect faults in actuators and sensors. If a fault is detected, the structure of the controller can be changed to get the best possible response of the system, or the system can be stopped.

Fault-detection approaches can be classified as model-free and model-based paradigms [1], [2]. Model-free fault diagnosis includes all the techniques that do not rely upon models of the underlying system, while model-based methods try to diagnose faults using the redundancy of some mathematical description of the dynamics. Examples of model-free techniques are the methods based on spectral analysis, pattern recognition and statistical classification, and the classical limit and trend check [3]. From the beginning of the seventies, there have been numerous theoretical advancements in fault diagnostics based on analytical redundancy. According to this approach, all the information on the system can be used to monitor the behaviour of the plant, including the knowledge about the dynamics. The presence of faults is detected by means of the so-called residuals, i.e., quantities that are over-sensitive to the malfunctions. Residual generation can be performed in different ways: parity equations [4], observer-based generation [5], and the methods based on parameter estimation [6]. Neural networks and fuzzy systems have also been applied in model-based FDI [7]. The necessity to obtain good diagnostic performances without installing a lot of redundant and dedicated expensive equipment, forces the diagnostic tools developers the techniques available to processing all the information that are "hidden" in the technological process. In fact what the reality of industrial systems can offer to the engineer charged to implement the monitoring functions, is usually very inadequate: poor models available, lack of redundancy, insufficient number of measures, noise on the acquired data, unmodeled disturbances [8], [9] and [10].

All failure detection methods exploit redundant data which is obtained either directly, when two or more sensors are available for measurement of a process variable, or analytically, when a process variable is estimated using the mathematical process model. These redundancy relationships may then be exploited to generate residual signals. Under normal operating conditions these residuals are "small" in an appropiate sense and yet display distinct patterns when failures occur. The failure diagnosis process consists in three stages:

a) *Modelling of process* (state estimation, parameter estimation, statistical decision theory, etc.).

b) *Residual generation*. The residuals are independent of true measurements but reflect the effects of modeling uncertainty, noise and component failures. In the absence of failures and gross modeling errors the residuals are unbiased showing agreement between measurements and model-based predictions.

c) *Residual analysis*. Due to the effects of noise and model uncertainty the residuals must be carefully examined to determine the presense of failures (detection) and which system components have failed (isolation).

2. THE GENERATION OF RESIDUE DURING THE MODEL-BASED DIAGNOSIS

When the system has the actuators affected by faults, this situation can be described by the relation:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Bu_d(t) \\ y(t) = Cx(t) + Du(t) + Du_d(t) \end{cases} \qquad (1)$$

where $x(t)$ is the system's state, $y(t)$ is the system's output, $u(t)$ is the system's command and the vector $u_d(t)$ represent the fault vectors for the actuators. The transfer function type input-output representation for the system is the following:

$$y(s) = H(s)u(s) + H(s)u_d(s) \qquad (2)$$

where

$$H(s) = C(sI - A)^{-1}B + D \qquad (3)$$

The residue generator is a linear processor whose inputs consist in the input and output of the monitored system. This structure can be expressed mathematically so (Figure 1):

$$r(s) = \left[P(s)Q(s)\right]\cdot\begin{bmatrix} u(s) \\ y(s) \end{bmatrix} = P(s)u(s) + Q(s)y(s) \qquad (4)$$

The matrixes $P(s)$ and $Q(s)$ are transfer matrixes built using linear, stabile systems. According to the definition, the residue is designed to become 0 in the case of fault absence and different from 0, in the presence of faults.

$$r(t) = 0 \text{ if and only if } u_d(t) = 0 \qquad (5)$$

For the residue generator $r(s)$ to be a fault indicator, the transfer matrixes $P(s)$ and $Q(s)$ must satisfy the relation:

$$P(s) + Q(s)H(s) = 0 \qquad (6)$$

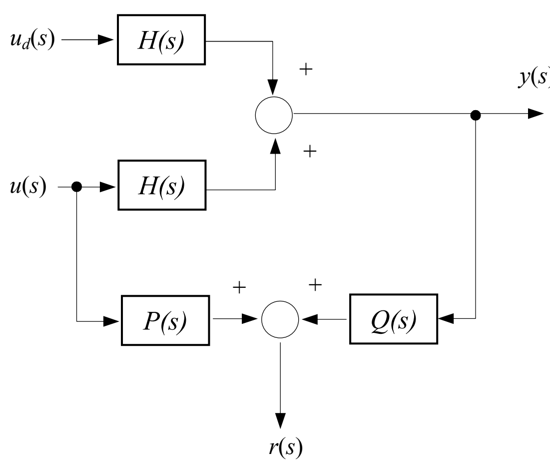

Fig. 1. The generation of the residue.

3. RECONFIGURATION PROBLEM

The goal of reconfiguration is to keep the reconfigured control loop performance sufficient for preventing plant shutdown. The following goals are distinguished:

- Stabilisation
- Equilibrium recovery
- Output trajectory recovery
- State trajectory recovery

Internal stability of the reconfigured closed loop is usually the minimum requirement. The equilibrium recovery goal (also referred to as weak goal) refers to the steady-state output equlibrium which the reconfigured loop reaches after a given constant input. This equilibrium must equal the nominal equilibrium under the same input (as time tends to infinity). This goal ensures steady-state reference tracking after reconfiguration.

The output trajectory recovery goal (also referred to as strong goal) is even stricter. It requires that the dynamic response to an input must equal the nominal response at all times. Further restrictions are imposed by the state trajectory recovery goal, which requires that the state trajectory be restored to the nominal case by the reconfiguration under any input.
Usually a combination of goals is pursued in practice, such as the equilibrium recovery goal with stability.

4. FAULT MODELLING

Faults may be represented as unknown inputs acting on the system (*additive faults*) or as changes of some plant parameters (*multiplicative faults*). A classification of various types of faults, generally accepted by specialists, is represented in Table 1 made by J. Gertler in 1998.

Table 1: A classification of faults [11].

	Additive	Multiplicative
Faults	Sensor fault Actuator fault	Parametric fault (plant fault)
Disturbances	Plant disturbances	Modeling error
Noises	Sensor noise Actuator noise Plant noise	

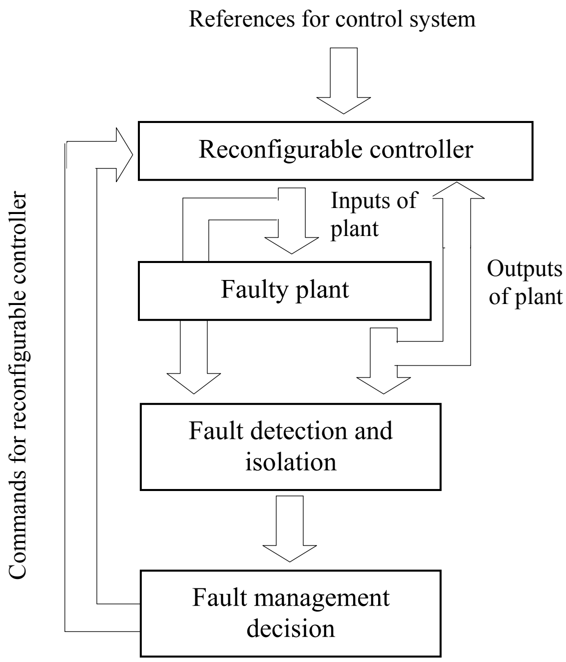

Fig. 2. The structure of the system with reconfigurable controller.

The Figure 2 shows a plant controlled by a controller in a standard control loop. The nominal linear model of the plant is

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{7}$$

The plant subject to a fault is modeled in general by

$$\begin{cases} \dot{x}_F(t) = C_F x_F(t) + B_F u(t) \\ y_F(t) = C_F x_F(t) \end{cases} \tag{8}$$

where the subscript F indicates that the system is faulty. This approach models multiplicative faults by modified system matrices. Specifically, actuator faults are represented by the new input matrix B_F, sensor faults are represented by the output matrix C_F, and internal plant faults are represented by the system matrix A_F. The bottom part of the Figure 2 shows a supervisory loop consisting of *fault detection and isolation* (FDI) and *reconfiguration* which changes the loop by:

- choosing new input and output signals to reach the control goal,
- changing the controller internals (including dynamic structure and parameters),
- adjusting the reference input.

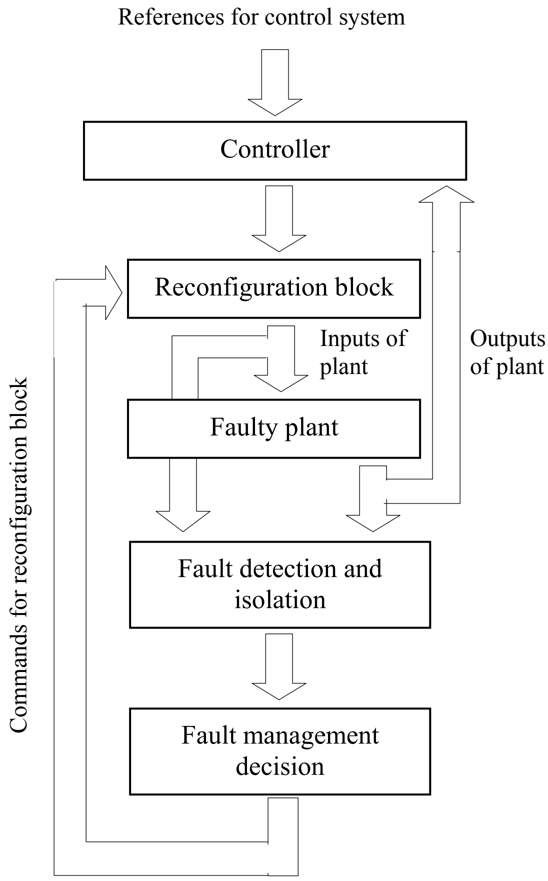

Fig. 3. The structure of the system which assures the fault hiding. The goals of the control system and the nominal controller are preserved.

5. FAULT HIDING

During the following the fault hiding principle is presented. A reconfiguration block is placed between faulty plant and nominal controller (Fig. 3). The reconfigured plant behaviour must match the nominal behaviour. Furthermore, the reconfiguration goals are pointed out.
This paradigm aims at keeping the nominal controller in the loop. To this end, a reconfiguration block is placed between the faulty plant and the nominal controller. Together with the faulty plant, it forms the reconfigured plant. The reconfiguration block has to fulfill the requirement that the behaviour of the reconfigured plant matches the behaviour of the nominal that is fault-free plant.

6. APPLICATION

To exemplify the algorithm which assures the hiding of the fault, let's consider a MIMO process with two inputs and two outputs.

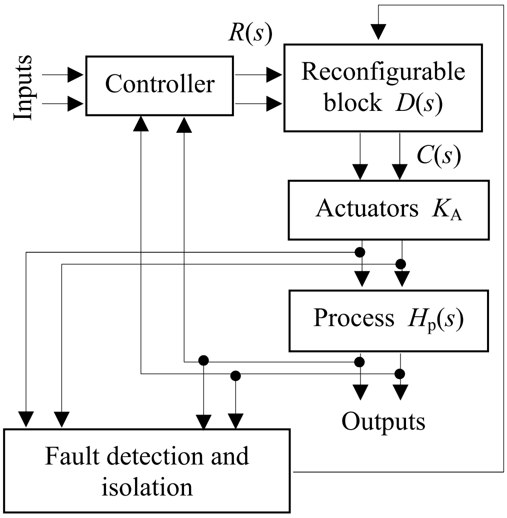

Fig. 4. The structure of reconfigurable system.

The structure of the reconfigurable system is presented in Fig. 4.

First, let's consider the normal situation (fault free). Let be the matrix transfer attached to the process:

$$H_p(s) = \begin{bmatrix} H_{p11}(s) & H_{p12}(s) \\ H_{p21}(s) & H_{p22}(s) \end{bmatrix} \tag{9}$$

and the matrix transfer function for actuators, who are considered proportionally, for simplify the calculus:

$$K_A = \begin{bmatrix} k_{a1} & 0 \\ 0 & k_{a2} \end{bmatrix} \tag{10}$$

The transfer matrix for reconfigurable block is:

$$D(s) = \begin{bmatrix} D_{11}(s) & D_{12}(s) \\ D_{21}(s) & D_{22}(s) \end{bmatrix} \tag{11}$$

where:

$$D_{ij}(s) = \left. \frac{C_i(s)}{R_j(s)} \right|_{R_k(s)\equiv 0, \forall k \neq j, i=\overline{1,2}, j=\overline{1,2}, k=\overline{1,2}.} \tag{12}$$

and where the $R_j(s)$ are the outputs of the controller and the $C_i(s)$ are the outputs of the reconfigurable block (the commands for the actuators).

To obtain a faster response of the system, we can choose $D(s)$ for instance, with the form:

$$D(s)=\begin{bmatrix} 1 & D_{12}(s) \\ D_{21}(s) & 1 \end{bmatrix} \quad (13)$$

For carry out the uncoupling, it is necessary that the transfer matrix:

$$G(s)=H_p(s)K_A D(s) \quad (14)$$

to has the form:

$$G(s)=\begin{bmatrix} G_{11}(s) & 0 \\ 0 & G_{22}(s) \end{bmatrix} \quad (15)$$

So, we have:

$$\begin{aligned} &k_{a1}H_{p11}(s)D_{12}(s)+k_{a2}H_{p12}(s)=0 \\ &\Rightarrow D_{12}(s)=-\frac{k_{a2}H_{p12}(s)}{k_{a1}H_{p11}(s)} \end{aligned} \quad (16)$$

$$\begin{aligned} &k_{a2}H_{p22}(s)D_{21}(s)+k_{a1}H_{p21}(s)=0 \\ &\Rightarrow D_{21}(s)=-\frac{k_{a1}H_{p21}(s)}{k_{a2}H_{p22}(s)} \end{aligned} \quad (17)$$

and respectively:

$$G_{11}(s)=k_{a1}\left[H_{p11}(s)-\frac{H_{p12(s)}H_{p21}(s)}{H_{p22}(s)}\right] \quad (18)$$

$$G_{22}(s)=k_{a2}\left[H_{p22}(s)-\frac{H_{p12(s)}H_{p21}(s)}{H_{p11}(s)}\right] \quad (19)$$

Another suggestion of the authors, if it is possible, is to impose the $G(s)$ matrix with the form:

$$G(s)=\begin{bmatrix} H_{p11}(s) & 0 \\ 0 & H_{p22}(s) \end{bmatrix} \quad (20)$$

In this case we respect the dynamic input-output of the process on the principals channels. In these conditions, for $D(s)$ with generally form (11), we have the relations:

$$\begin{aligned} &k_{a1}H_{p11}(s)D_{11}(s)+k_{a2}H_{p12}(s)D_{21}(s)= \\ &=H_{p11}(s)=G_{11}(s) \end{aligned} \quad (21)$$

$$k_{a1}H_{p11}(s)D_{12}(s)+k_{a2}H_{p12}(s)D_{22}(s)=0 \quad (22)$$

$$k_{a1}H_{p21}(s)D_{11}(s)+k_{a2}H_{p22}(s)D_{21}(s)=0 \quad (23)$$

$$\begin{aligned} &k_{a1}H_{p21}(s)D_{12}(s)+k_{a2}H_{p22}(s)D_{22}(s)= \\ &=H_{p22}(s)=G_{22}(s) \end{aligned} \quad (24)$$

From (21), (22), (23) and (24) we obtain the expression of the components of $D(s)$ matrix:

$$\begin{aligned} &D_{11}(s)=[H_{p11}(s)H_{p22}(s)]\times \\ &\times\frac{1}{k_{a1}[H_{p11}(s)H_{p22}(s)-H_{p12}(s)H_{p21}(s)]} \end{aligned} \quad (25)$$

$$\begin{aligned} &D_{12}(s)=-[H_{p12}(s)H_{p22}(s)]\times \\ &\times\frac{1}{k_{a1}[H_{p11}(s)H_{p22}(s)-H_{p12}(s)H_{p21}(s)]} \end{aligned} \quad (26)$$

$$\begin{aligned} &D_{21}(s)=-[H_{p11}(s)H_{p21}(s)]\times \\ &\times\frac{1}{k_{a2}[H_{p11}(s)H_{p22}(s)-H_{p12}(s)H_{p21}(s)]} \end{aligned} \quad (27)$$

$$\begin{aligned} &D_{22}(s)=[H_{p11}(s)H_{p22}(s)]\times \\ &\times\frac{1}{k_{a2}[H_{p11}(s)H_{p22}(s)-H_{p12}(s)H_{p21}(s)]} \end{aligned} \quad (28)$$

If we consider also the transfer matrix attached to the controllers:

$$C(s)=\begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix} \quad (29)$$

the transfer matrix for the structure represented in Fig. 4 is:

$$S(s)=\begin{bmatrix} \dfrac{G_{11}(s)C_1(s)}{1+G_{11}(s)C_1(s)} & 0 \\ 0 & \dfrac{G_{22}(s)C_2(s)}{1+G_{22}(s)C_2(s)} \end{bmatrix} \quad (30)$$

If

$$S(s)\equiv S^*(s)=\begin{bmatrix} S_{11}^*(s) & S_{12}^*(s) \\ S_{21}^*(s) & S_{22}^*(s) \end{bmatrix} \quad (31)$$

is a desired matrix by the designer, we obtain for the controllers the next expressions:

$$C_i(s)=\frac{S_{ii}^*(s)}{1-S_{ii}^*(s)}\frac{1}{G_{ii}(s)},i=\overline{1,2}. \quad (32)$$

If the process has been modified (internal faults) it will have a modified transfer $H_{pF}(s)$ matrix. It is *fault detection and isolation* system's duty to calculate that.

The objective of the method is to maintain the controller unmodified. In order to do this, the reconfigurable block must adapt to the new conditions. This means that the $S(s)$ and $G(s)$ matrix must preserve their initial forms, so the controller remains unmodified. From (21), (22), (23) and (24) we obtain the expression of the components of $D_F(s)$ matrix in the fault condition:

$$D_{F11}(s) = H_{p11}(s)H_{pF22}(s)\times \times\frac{1}{k_{a1}[H_{pF11}(s)H_{pF22}(s)-H_{pF12}(s)H_{pF21}(s)]} \quad (33)$$

$$D_{F12}(s) = -H_{pF12}(s)H_{p22}(s)\times \times\frac{1}{k_{a1}[H_{pF11}(s)H_{pF22}(s)-H_{pF12}(s)H_{pF21}(s)]} \quad (34)$$

$$D_{F21}(s) = -H_{p11}(s)H_{pF21}(s)\times \times\frac{1}{k_{a2}[H_{pF11}(s)H_{pF22}(s)-H_{pF12}(s)H_{pF21}(s)]} \quad (35)$$

$$D_{F22}(s) = H_{pF11}(s)H_{p22}(s)\times \times\frac{1}{k_{a2}[H_{pF11}(s)H_{pF22}(s)-H_{pF12}(s)H_{pF21}(s)]} \quad (36)$$

7. CONCLUSIONS

The presented algorithm is very useful when the parameters of the process are affected by small modifications, that influence the functions of the system in a slightly degraded manner.

The method has its limitations. If the faulty plant suffers significant changes, which affect the entire structure, it is difficult, if not impossible, to maintain the same controller.

The association between the *fault detection and isolation* and the *reconfiguration methods* can assure a certain immunity from the faults of the process. From this point of view, the method is closer to the robust control algorithm.

REFERENCES

[1]Frank, P.M., Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. A survey and some new results. Automatica 26(3), p. 459-474, 1990.

[2]Gertler, J., Survey of model-based failure detection and isolation in complex plants. IEEE Control Systems Mag. 9, p. 3-11, 1988.

[3] Pau, L.F., Failure diagnosis and performance monitoring, Marcel Dekker, New York, 1981.

[4]Gertler, J., Fault detection and isolation using parity relations. Control Engineering Practice, 5, p. 653-661, 1997.

[5]Patton, R.J. and J. Chen, Observer-based fault detection and isolation: robustness and applications. Control Engineering Practice, 5, p. 671-682, 1997.

[6]Isermann, R., Process fault detection based on modelling and estimation methods-a survey. Automatica, 20(4), p. 387-404, 1984.

[7]Patton, R.J., F J Uppal and C J Lopez-Toribio, Soft Computing Approaches to Fault Diagnosis for Dynamic Systems: A Survey. 4th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, Budapest, 14-16 June 2000, 1, p. 298-311, 2000.

[8]Isermann, R., Supervision, fault-detection and fault-diagnosis methods. An introduction. Control Engineering Practice, 5, p. 639-652, 1997.

[9]Isermann, R. and P. Ballé, Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice, 5, p. 709-719, 1997.

[10]Patton, R.J., Fault-tolerant control systems: the 1997 situation. IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, Kingston, UK, 3, p. 1033-1054, 1997.

[11]Gertler, J., Fault Detection and Diagnosis in Engineering Systems, Marcel Dekker, New York, p. 155, 1998.