



HAL
open science

An algorithm for obtaining the redundancy equations of LTI systems

José Ragot, Didier Maquin

► **To cite this version:**

José Ragot, Didier Maquin. An algorithm for obtaining the redundancy equations of LTI systems. Automatica, 1994, 30 (3), pp.537-542. 10.1016/0005-1098(94)90133-3 . hal-00144004

HAL Id: hal-00144004

<https://hal.science/hal-00144004>

Submitted on 29 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**AN ALGORITHM FOR OBTAINING
THE REDUNDANCY EQUATIONS OF LTI SYSTEMS**

José RAGOT[†] and Didier MAQUIN

Centre de Recherche en Automatique de Nancy - CNRS UA 821

BP 40 - Rue du Doyen Marcel Roubault

54 501 Vandoeuvre Cedex, FRANCE

Tél : (33) 83 50 30 80

Fax : (33) 83 50 30 96

[†] Author to whom correspondence and reprints should be sent

AN ALGORITHM FOR OBTAINING THE REDUNDANCY EQUATIONS OF LTI SYSTEMS

KEY WORDS

Analytical redundancy, failure detection, diagnosis, parity equations, linear systems.

ABSTRACT

In the context of model-based failure detection and diagnosis, there is residual generation which can be obtained through a generalized parity space concept. This paper presents a new algorithm involving the generation of redundancy or parity equations for dynamic linear time-invariant systems. The proposed scheme involves the successive elimination of state sub-vectors and their replacement by output values as well as the processing of an output done one at a time to avoid matrix inversion. Some examples point out the numerical efficiency of the proposed procedure.

INTRODUCTION

A large variety of techniques have been proposed in the last decade for the detection, the isolation and the estimation of faults in dynamic systems (Chow and Willsky, 1984), (Iserman, 1984), (Frank, 1990). According to the general terminology, these techniques are related to the following tasks :

- event detection which probably indicates that something is going wrong in the process,
- isolation which is concerned with the exact localization of the events,
- identification and characterization of the events in order to estimate their magnitudes and to give the final diagnosis on the functioning of the process.

The usual procedure of fault detection can be roughly divided into the following two steps : generation of a so-called residuals (functions that are accentuated by the faults) and fault detection based on a logical device (in order to monitor both the time of occurrence and the location of the faults). Whatever the techniques of fault detection, they all need the generation of signals that are accentuated by the presence of faults. These signals must be close to zero when the process behavior is consistent with the model and as different as possible from zero when this is not the case (moreover, if it is possible, they must not be altered by the causes of false alarm). The procedures for generating these signals are numerous but may be classified into two groups : input-output residuals obtained from the process model and residual-like quantity resulting from the identification of the process model parameters. In each case supplementary information is provided by the calculation

of the covariance matrix of the residuals (this is needed by the decision device in order to compare the weights of the different residuals) ; often the covariance matrix is calculated recursively which allows a comparison of the consecutive values of the residuals without referring to a prespecified value.

The work described here only focuses on model-based redundancy. As discussed in the papers of Deckert (1977), Chow (1984), Massoumnia (1988) these redundancies are simply functions of the time evolution of the process : the residual generator takes, as inputs, the commands of the actuators and the sensor outputs and points out a set of residuals which are monitored for failure detection. Assuming that the faults are detectable, four requirements are mainly formulated to obtain a powerful failure detection algorithm. First, the residuals must be independent of the state of the system : they are only due to noise in the absence of failures. Second, the failures have to be distinguished from each other ; this is the problem of isolation which results in a correct choice of the coding equations strategy. This means that, in an ideal situation, any one sensor affects just one residual and the affected residual can be associated with only one sensor. Thirdly, the detection scheme has to be insensitive to uncertainties affecting the system (for example, modification of its transfer function, unknown inputs or modeling errors) ; this is the general property of robustness (see for example (Lou, 1986) and (Wünnenberg, 1989)) which is necessary in order to achieve reliability (i.e. to minimize false alarms and missed failure detections). Lastly, the observation window used for the residual calculation has to be as short as possible in order to have a rapid failure detection. The last two requirements are often conflicting, as, for example, an important duration of observation improves the detection robustness but increases the fault detection delay.

Let us now return to the residual generation through model equations. Several principles may be generally applied among which are : Kalman filtering, direct input-output model relations, observers or parameter identification. Note that in the fields of chemical and metallurgical engineering and also for power plant processes, special attention has been given to material and energy balance equations ; these are used to generate residuals from the direct relation between the different measured flowrates (material or energy). Even if these relations are not general, they have the considerable advantage of being structurally exact because they express the laws of mass and energy conservation ; consequently, if the residuals differ from zero, this deviation is only due to the faults. For the static case, Vaclavek (1974) and Serth (1986) proposed several tests for balance equation analysis which have since been extended to the multilinear case ; at the same time the dynamic balance equations have been examined (Gertler, 1973). A popular approach which is directly issued from the input-output description of a process is known as parity equations (Potter, 1977) which is available for static as well as dynamic processes (Willksy, 1976

and Chow, 1984)). The second well-known approach is based on the reconstruction of the output of the process with the aid of observers either in a deterministic or stochastic situation ; it uses the estimation error or a function of it as a residual (Clark, 1978). Both mentioned approaches are different ways for solving the same kind of problem. Moreover, recent papers state the equivalence between residual generation by observer and parity space approaches which was first pointed out by Frank (1990) ; this equivalence problem has been recently addressed in the following papers : (Gertler, 1991), (Staroswiecki, 1991), (Patton, 1991) and (Magni, 1991).

This paper focuses on the generation of residuals from the dynamic input-output equations of processes ; we do not discuss the outcomes of the statistical testing of these residuals which can be found for example in Patton (1989). An introductory example is given to point out the problem of redundancy and the principle of redundancy extraction. We then suggest a new algorithm which directly extracts the redundant equations from the model process without involving a preliminary state or output estimation of the process. The proposed scheme involves a successive elimination of state sub-vectors and their replacement by measured output values as well as the processing of one output at a time to completely avoid matrix inversion.

INTRODUCTORY EXAMPLE

To illustrate the idea of redundancy generation, consider a simple dynamic system ideally modeled by the discrete third order state equations :

$$x_1(k+1) = 0.5 x_1(k) - 0.7 x_2(k) + 0.7 x_3(k) \quad (1a)$$

$$x_2(k+1) = 0.1 x_1(k) + 0.8 x_2(k) + 0.1 x_3(k) + u(k) \quad (1b)$$

$$x_3(k+1) = -x_1(k) \quad (1c)$$

$$y(k) = x_3(k) \quad (1d)$$

In these equations, only a function of the state variables is measured and the input $u(k)$ is assumed to be known. The obtaining of redundancy equations (if they exist) consists of eliminating the unknown variables x_1 , x_2 , x_3 between the state equations. For this example, direct elimination is straightforward and gives the equation :

$$y(k+3) - 1.3 y(k+2) + 1.17 y(k+1) - 0.63 y(k) - 0.7 u(k) = 0 \quad (2)$$

For this redundancy relation, there are different possible methods for the residual generation. The redundancy equation itself can be taken as a residual. For our example, this would be :

$$r_1(k) = y(k) - 1.3 y(k-1) + 1.17 y(k-2) - 0.63 y(k-3) - 0.7 u(k-3)$$

Such a residual is a moving average process, since it is a function of the most recent sensor output and input values. If the sensor is subject to a failure, the residual value $r(k)$ is correlated with $r(k-1)$, $r(k-2)$ and $r(k-3)$ but not with any of its values preceding $r(k-3)$.

A second possibility is to solve (2) recursively and to compare the result each time to the actual output of the sensor. Therefore, we calculate :

$$\hat{y}(k) = 1.3 \hat{y}(k-1) - 1.17 \hat{y}(k-2) + 0.63 \hat{y}(k-3) + 0.7 u(k-3)$$

and the resulting residual is :

$$r_2(k) = y(k) - \hat{y}(k)$$

It should be noticed that the residual $r_2(k)$ is not a moving average of the previous values as it involves the integration of $u(k)$. For example, a bias on the actuator will not be removed in a finite number of steps.

A third method of residual generation is also based on the model of $\hat{y}(k)$ but explicitly takes into account noise affecting the process ; as a result, we can design a Kalman filter from which the innovation sequence directly gives the residuals.

Whatever the methods used to generate the residuals, these latter are representative of the faults to be detected. All the methods that may be applied to analyze the residuals use the principle of change detection ; these changes look like abrupt modifications of the residual, slow shifts, or abnormal values. A survey of the different techniques may be found in Basseville (1986) for change detection in signals and systems and Barnett (1984) for outlier detection.

GENERAL APPROACH

From a more general point of view, consider the discrete linear time-invariant system :

$$x(k+1) = A x(k) + B u(k) \tag{3a}$$

$$y(k) = C x(k) \tag{3b}$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^1$ and appropriate dimensions for the matrices A, B and C. Using z as the shift forward operator ($z f(k) = f(k+1)$), the elimination of the state x gives the redundancy equation :

$$y(k) = C (z I - A)^{-1} B u(k) \tag{4}$$

This redundancy equation may also be expressed :

$$H(z) y(k) = G(z) u(k) \tag{5}$$

where the matrices $G(z)$ and $H(z)$ are related to those of the state model :

$$G(z) = C \text{Adj}(z I - A) B \quad (6a)$$

$$H(z) = \text{Det}(z I - A) \quad (6b)$$

Applying equation (5) to the available measurements on y and u yields a set of parity equations where $e(k)$ is in general not zero :

$$e(k) = H(z) y(k) - G(z) u(k) \quad (7)$$

The deviation from zero is the combined result of measurement noises and faults ; if the noise is negligible the residual may be directly analyzed, otherwise a preliminary statistical treatment must be applied on $e(k)$. The parity vector $e(k)$ can be readily calculated if $u(k)$ and $y(k)$ are known ; the columns of the matrix $(H(z) \ -G(z))$ are known as failure directions and the vector space spanned by these directions is referred to as the parity space in which the parity vector lies. The detectability and the isolability of the faults is then strongly connected to the structure of the matrix $(H(z) \ -G(z))$. A failure may be detected by comparing the norm of $e(k)$ to a specified threshold ; the localization of the failure is then performed by comparing the direction of the vector $e(k)$ to the failure directions. In either case a pattern recognition procedure is necessary to isolate failure through their signatures.

A direct formulation of the transfer function may also be obtained by the stable factorization approach (Kailath 1980, Vidyasagar 1985). Given any proper rational transfer function matrix $P(z)$, we can define the stable right and left coprime factorization as :

$$P(z) = N(z) D^{-1}(z) \quad (8a)$$

$$P(z) = \bar{D}^{-1}(z) \bar{N}(z) \quad (8b)$$

where the different matrices N, D, \bar{N} and \bar{D} verify :

$$\begin{pmatrix} Y & X \\ -\bar{N} & \bar{D} \end{pmatrix} \begin{pmatrix} D & -\bar{X} \\ N & \bar{Y} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \quad (9)$$

These left and right stable factors (N, D) and (\bar{N}, \bar{D}) may also be obtained from state-space descriptions of $P(z)$ for the system described by equations (3) where the transfer function is expressed by :

$$P(z) = C (z I - A)^{-1} B \quad (10)$$

Stabilizability and detectability ensure that K and F can be found such that $A-BK$ and $A-FC$ are stable matrices. Then we have (Nett, 1984) :

$$\begin{pmatrix} Y & X \\ -\bar{N} & \bar{D} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + \begin{pmatrix} K \\ -C \end{pmatrix} (zI - A + FC)^{-1} (B \quad F) \quad (11a)$$

$$\begin{pmatrix} D & -\bar{X} \\ N & \bar{Y} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + \begin{pmatrix} K \\ -C \end{pmatrix} (zI - A + BK)^{-1} (B \quad F) \quad (11b)$$

With a judicious choice of the matrices F and K it is possible to reduce the calculation work which allows the determination of the matrices N , D , \bar{N} and \bar{D} . This has been suggested by (Viswanadham, 1987) in a particular case of the state equations structure ; a more general method is given in (Fang, 1991). See also (Ding, 1990) and (Frank, 1991) for the application of the factorization applied to fault detection ; it was demonstrated that all residual generators can be parametrized in the form :

$$r(z) = Q(z) (\bar{D}(z) y(z) - \bar{N}(z) u(z)) \quad (12)$$

where $Q(z)$ denotes the parameter matrix chosen such that some specifications about fault isolation are satisfied (structured residual generation).

Despite the simplicity of the formulation, the generation of the redundancies is achieved through matrix calculations involving inverse computations (equations (6a) and (6b) for the direct generation or equations (11a) and (11b) when using the coprime factorization). Faddeev (1963) suggested an elegant algorithm in order to compute the inverse of the matrix $(zI - A)$. Bingulac (1975) proposed a method which allows the computation of the transfer functions $(H(z), G(z))$ without the evaluation of the cofactor matrix. Blackwell (1984) and Hashim (1988) formulated the same problem through a triangularisation procedure of the state equations. Varga (1981) proposed a calculation of the transfer function between each input and each output involving an orthogonal transformation which eliminates the noncontrollable and nonobservable parts of each transfer function. Furthermore, Misra (1987) suggested an approach using the lower and upper Hessenberg form for each input-output pair, the realization of which being totally controllable and observable. In the extended case of singular systems, Paraskevopoulos (1984) developed two methods using the pencil matrix properties and the drazin inverse matrices ; this latter point was also examined by Saidahmed (1990). Another approach is based on the division of the state equations using the concept of observability but it also needs matrix calculation such as rank of matrix.

AN EXAMPLE OF REDUNDANCY GENERATION

Here, we suggest an algorithm based on elimination of the variables. Let us remember that the basic principle for the redundancy equation generation consists of eliminating the unknown variables. This elimination may be performed by aggregation of equations (linear combinations) or by projection. To begin with, we reformulate the state equations used for the previous example. Thus we try to eliminate the states x_1 , x_2 et x_3 . Considering x_3 as the measured part of the state vector, it is possible to rewrite the equations (1) as :

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} 0.5 & -0.7 \\ 0.1 & 0.8 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} 0.7 & 0.0 \\ 0.1 & 1.0 \end{pmatrix} \begin{pmatrix} y(k) \\ u(k) \end{pmatrix} \quad (13a)$$

$$y(k+1) = (-1 \quad 0) \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \quad (13b)$$

One then obtains a dynamic state equation of reduced dimension and a measurement equation which only depend on two unknown state variables. Once again, it is possible to isolate the "measured" component x_1 . This leads to the reduced state equation :

$$x_2(k+1) = 0.8 x_2(k) + (-0.1 \quad 0.1 \quad 1) \begin{pmatrix} y(k+1) \\ y(k) \\ u(k) \end{pmatrix} \quad (14a)$$

$$y(k+2) - 0.5 y(k+1) + 0.7 y(k) = 0.7 x_2(k) \quad (14b)$$

It is clear that equation (14a) describes the time evolution of x_2 due to the generalized input ($y(k+1)$, $y(k)$, $u(k)$) while equation (14b) corresponds to the new measurement equation. Moreover the elimination of the unknown state x_2 between the equation (14a) and (14b) is straightforward and gives the redundancy equation obtained before (equation 2).

GENERAL ALGORITHM

We have previously outlined the numerical difficulties of the direct approach of the redundancy equation generation ; this is mainly due to the dimension of the matrices to be inverted. Our objective is to avoid these matrix inversions. When using successive transformations one may reduce the state equations of the system to one scalar equation ; generation of redundancy equations is then straightforward. The proposed scheme, for the previous example, uses a progressive reduction of the dimension of the state equations ; each step of this reduction is characterized by the suppression of one state variable. The method is continued until a constant or a matrix of minimal dimension is reached : in the previous example when the dimension of the state equation is equal to one. This is the basic idea of our strategy of redundancy equation generation : the unknown variables x are

eliminated from the state equations in order to keep only the known variables u and y . The previous example states that this elimination is achieved without calculating the inverse of the matrix $(zI - A)$. Let us return to the general state equations (3). With C_1 a non singular part of the matrix C , a simple permutation of the components of x enables the division :

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u(k) \quad (15a)$$

$$y(k) = C_1 x_1(k) + C_2 x_2(k) \quad (15b)$$

where $x_1(k) \in \mathbb{R}^l$, $x_2(k) \in \mathbb{R}^{n-l}$ and the matrices have appropriate dimensions. This division assumes that the C matrix has full row rank ; if this hypothesis does not hold (some measures are then physically redundant), one may reduce the dimension of the observation equations by considering only those which are independent.

With the invertible linear transformation :

$$\bar{x}_1(k) = C_1 x_1(k) + C_2 x_2(k) \quad (16a)$$

$$\bar{x}_2(k) = x_2(k) \quad (16b)$$

the state equations are rewritten :

$$\begin{pmatrix} \bar{x}_1(k+1) \\ \bar{x}_2(k+1) \end{pmatrix} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} \begin{pmatrix} \bar{x}_1(k) \\ \bar{x}_2(k) \end{pmatrix} + \begin{pmatrix} \bar{B}_1 \\ \bar{B}_2 \end{pmatrix} u(k) \quad (17a)$$

$$y(k) = \bar{x}_1(k) \quad (17b)$$

with the definitions :

$$\bar{A}_{11} = (C_1 A_{11} + C_2 A_{21}) C_1^{-1} \quad (18a)$$

$$\bar{A}_{12} = -\bar{A}_{11} C_2 + C_1 A_{12} + C_2 A_{22} \quad (18b)$$

$$\bar{A}_{21} = A_{21} C_1^{-1} \quad (18c)$$

$$\bar{A}_{22} = A_{22} - A_{21} C_1^{-1} C_2 \quad (18d)$$

$$\bar{B}_1 = C_1 B_1 + C_2 B_2 \quad (18e)$$

$$\bar{B}_2 = B_2 \quad (18f)$$

Then, the elimination of $\bar{x}_1(k)$ and $\bar{x}_2(k)$ between equations (17a) and (17b) gives the redundancy equation :

$$[(zI - \bar{A}_{11}) - \bar{A}_{12} (zI - \bar{A}_{22})^{-1} \bar{A}_{21}] \bar{X}_1(z) - [\bar{B}_1 + \bar{A}_{12} (zI - \bar{A}_{22})^{-1} \bar{B}_2] U(z) = 0 \quad (19)$$

It appears that this form is more advantageous than (4) considering the size of the matrix to be inverted. However, the size of this remaining matrix may be too large. A more interesting presentation of the equation can now be obtained by eliminating the variable $\bar{x}_1(k)$ in the state equation (17) ; the remaining equations are reduced to the part containing the unknown state variables $\bar{x}_2(k)$:

$$\bar{x}_2(k+1) = \bar{A}_{22} \bar{x}_2(k) + (\bar{A}_{21} \quad \bar{B}_2) \begin{pmatrix} y(k) \\ u(k) \end{pmatrix} \quad (20a)$$

$$z(k) = \bar{A}_{12} \bar{x}_2(k) \quad (20b)$$

$$\text{with } z(k) = y(k+1) - \bar{A}_{11} y(k) - \bar{B}_1 u(k) \quad (20c)$$

This form points out the generalized input ($y(k)$, $u(k)$) which drives the evolution of the state variable $\bar{x}_2(k)$ and the generalized measurement $z(k)$. Consequently, equations (20) are structurally identical to equations (3) ; therefore, the transformation used in equations (15) and (16) may be applied to equation (20). In this way, we eliminate the unknown variables $\bar{x}_2(k)$. The complete solution will now be presented ; as mentioned above, the algorithm is applied sequentially and each step is referenced by a supplementary index "n". For the first step, $n=0$, the state equations (3) are rewritten :

$$x(0, k+1) = A(0) x(0, k) + B(0) u(0, k) \quad (21a)$$

$$y(0, k) = C(0) x(0, k) \quad (21b)$$

where, it is evident, at this step, that $x(0, k) = x(k)$ and the same simplification holds for the other variables and matrices. After the extraction of a regular part $C_1(0)$ of the matrix $C(0)$, we apply the previous results to obtain the partitioned state equations :

$$\begin{pmatrix} \bar{x}_1(0, k+1) \\ \bar{x}_2(0, k+1) \end{pmatrix} = \begin{pmatrix} \bar{A}_{11}(0) & \bar{A}_{12}(0) \\ \bar{A}_{21}(0) & \bar{A}_{22}(0) \end{pmatrix} \begin{pmatrix} \bar{x}_1(0, k) \\ \bar{x}_2(0, k) \end{pmatrix} + \begin{pmatrix} \bar{B}_1(0) \\ \bar{B}_2(0) \end{pmatrix} u(0, k) \quad (22a)$$

$$y(0, k) = \bar{x}_1(0, k) \quad (22b)$$

$$\text{with } \bar{x}_1(0, k) = C_1(0) x_1(0, k) + C_2(0) x_2(0, k)$$

$$\bar{x}_1(0, k) = x_2(0, k)$$

which also gives the reduced equations :

$$\bar{x}_2(0, k+1) = \bar{A}_{22}(0) \bar{x}_2(0, k) + (\bar{A}_{21}(0) \quad \bar{B}_2(0)) \begin{pmatrix} y(0, k) \\ u(0, k) \end{pmatrix} \quad (23a)$$

$$\bar{y}(0, k) = \bar{A}_{12}(0) \bar{x}_2(0, k) \quad (23b)$$

with the pseudo-measurement :

$$\bar{y}(0, k) = \bar{x}_1(0, k+1) - \bar{A}_{11}(0) \bar{x}_1(0, k) - \bar{B}_1 u(0, k) \quad (23c)$$

The reader will notice that $\bar{y}(0, k)$ depends on a signal at time $(k+1)$; as we are only looking for equation generation, no problems of time synchronization have to be taken into account. In order to simplify the notations, we will drop the subscript 2 of the state variable by defining the state variable at step 1 with :

$$x(1, k) = \bar{x}_2(0, k) \quad (24)$$

So, at step 1, equations (23a) and (23b) are re-noted :

$$x(1, k+1) = A(1) x(1, k) + B(1) u(1, k) \quad (25a)$$

$$y(1, k) = C(1) x(1, k) \quad (25b)$$

with evident definitions for the new input $u(1, k)$ and measured variable $y(1, k)$ and also for the matrices $A(1)$, $B(1)$ and $C(1)$. More generally, the complete algorithm is stated as follows :

$$x(n, k+1) = A(n) x(n, k) + B(n) u(n, k) \quad n = 0, \dots, N \quad (26a)$$

$$y(n, k) = C(n) x(n, k) \quad n = 0, \dots, N \quad (26b)$$

with the pseudo measurement defined by :

$$y(n, k) = y(n-1, k+1) - \bar{A}_{11}(n-1) y(n-1, k) - \bar{B}_1(n-1) u(n-1, k) \quad (27c)$$

At the final step N , a first possibility occurs when the matrix A reduces to a scalar which allows the construction of the redundancy equation without matrix inversion ; eliminating $x(N, k)$ in equations (26a) and (26b) written for $n = N$ gives :

$$y(N, k) (z - A(N)) = C(N) B(N) u(N, k) \quad (28a)$$

The second possibility occurs when $C(N)$ becomes zero ; then, the pseudo-measurement $y(N, k)$ is zero and the redundancy equation must be generated from equation (20c) as :

$$y(N-1, k+1) - \bar{A}_{11}(N-1) y(N-1, k) - \bar{B}_1(N-1) u(N-1, k) = 0 \quad (28b)$$

Generalizing the transition from step 0 (equation (22)) to step 1 (equation (25)), at each step we have the following definitions :

$$A(n+1) = \bar{A}_{22}(n) \quad (29a)$$

$$\mathbf{B}(n+1) = (\bar{A}_{21}(n) \quad \bar{B}_2(n)) \quad (29b)$$

$$\mathbf{C}(n+1) = \bar{A}_{12}(n) \quad (29c)$$

$$\mathbf{A}(n) = \begin{pmatrix} A_{11}(n) & A_{12}(n) \\ A_{21}(n) & A_{22}(n) \end{pmatrix} \quad (29d)$$

$$\mathbf{B}(n) = \begin{pmatrix} B_1(n) \\ B_2(n) \end{pmatrix} \quad (29e)$$

$$\mathbf{C}(n) = (C_1(n) \quad C_2(n)) \quad (29f)$$

$$\bar{A}_{11}(n) = (C_1(n) A_{11}(n) + C_2(n) A_{21}(n)) C_1^{-1}(n) \quad (29g)$$

$$\bar{A}_{12}(n) = -\bar{A}_{11}(n) C_2(n) + C_1(n) A_{12}(n) + C_2(n) A_{22}(n) \quad (29h)$$

$$\bar{A}_{21}(n) = A_{21}(n) C_1^{-1}(n) \quad (29i)$$

$$\bar{A}_{22}(n) = A_{22}(n) - A_{21}(n) C_1^{-1}(n) C_2(n) \quad (29j)$$

$$\bar{B}_1(n) = C_1(n) B_1(n) + C_2(n) B_2(n) \quad (29k)$$

$$\bar{B}_2(n) = B_2(n) \quad (29l)$$

$$\mathbf{u}(n, k) = \begin{pmatrix} y(n-1, k) \\ u(n-1, k) \end{pmatrix} \quad (29m)$$

Note that the inverse of matrix $C_1(n)$ which appears in equation (22) may be simply avoided if we apply this algorithm for each output one after the other ; in this case, the matrix $C(n)$ reduces to a row-vector and the extraction of the non singular part is straightforward. The redundancy equations are then generated one output at a time and therefore can be considered as self-redundancy equations ; the isolation of the outputs in each equation may be used for the localization of the sensor failures.

A LAST SIMPLIFICATION

In the state equations obtained at each step, we have defined a generalized input vector (29m) and a pseudo-measurement vector (27c) ; these two vectors only depend on the input $u(k)$ and on the output $y(k)$ of the given system considered at different times. An appreciable simplification may be given to the previous algorithm by expressing these relations under a recursive form. At each step, the pseudo-measurement (27c) may be expressed :

$$y(n, k) = F(n-1, z) y(n-1, k) + G(n-1) u(n-1, k) \quad (30a)$$

where the matrices F and G are defined by :

$$F(n-1, z) = z - \bar{A}_{11}(n-1) \quad (30b)$$

$$G(n-1) = -\bar{B}_1(n-1) \quad (30c)$$

For the first step, it is clear that $y(n, k)$ and $u(n, k)$ only depend on the values of $y(0, k) = y(k)$ and $u(0, k) = u(k)$. According to the definitions (29m) and (30a), this dependence also appears at step n . We can make these relations clear at each step by defining :

$$y(n, k) = \begin{pmatrix} R(n, z) & S(n, z) \end{pmatrix} \begin{pmatrix} y(k) \\ u(k) \end{pmatrix} \quad (31a)$$

$$u(n, k) = \begin{pmatrix} P(n, z) & Q(n, z) \end{pmatrix} \begin{pmatrix} y(k) \\ u(k) \end{pmatrix} \quad (31b)$$

where R, S, P and Q are polynomial matrices of the shift variable z . To simplify the computation, one can then use a recursive formulation of these matrices. From (30a) written at step $(n+1)$ and using (31) we obtain :

$$y(n+1, k) = (F(n, z) R(n, z) + G(n) P(n, z)) y(k) + (F(n, z) S(n, z) + G(n) Q(n, z)) u(k) \quad (32)$$

Comparing (31a) written at step $(n+1)$ and (32), the following recurrences are deduced :

$$R(n+1, z) = F(n, z) R(n, z) + G(n) P(n, z) \quad (33a)$$

$$S(n+1, z) = F(n, z) S(n, z) + G(n) Q(n, z) \quad (33b)$$

From (29m) written at step $(n+1)$ and substituting the value of $y(n, k)$ and $u(n, k)$ given by equations (31) we obtain :

$$u(n+1, k) = \begin{pmatrix} R(n, z) & S(n, z) \\ P(n, z) & Q(n, z) \end{pmatrix} \begin{pmatrix} y(k) \\ u(k) \end{pmatrix} \quad (34)$$

Comparing (31b) and (34) the following recurrences are deduced :

$$P(n+1, z) = \begin{pmatrix} R(n, z) \\ P(n, z) \end{pmatrix} \quad (35a)$$

$$Q(n+1, z) = \begin{pmatrix} S(n, z) \\ Q(n, z) \end{pmatrix} \quad (35b)$$

The expressions of the pseudo-measurements and of the generalized input are deduced from the inputs and the outputs of the given system by using the equations (31a) and (31b); the matrices which define these transformations may be obtained recursively (33) and (35).

Finally, if in the equation (28a) we replace $y(N, k)$ and $u(N, k)$ by their respective definitions (31a) and (31b), we get the final form of the redundancy equations :

$$\begin{aligned} & (\mathbf{R}(N, z) (z - \mathbf{A}(N)) - \mathbf{C}(N) \mathbf{B}(N) \mathbf{P}(N, z)) \mathbf{y}(k) + \\ & (\mathbf{S}(N, z) (z - \mathbf{A}(N)) - \mathbf{C}(N) \mathbf{B}(N) \mathbf{Q}(N, z)) \mathbf{u}(k) = 0 \end{aligned} \quad (36a)$$

When (28b) has to be used we have the redundancy equation :

$$\mathbf{R}(N, z) \mathbf{y}(k) + \mathbf{S}(N, z) \mathbf{u}(k) = 0 \quad (36b)$$

EXAMPLE 1

Consider the four order system described by equations (3a) and (3b), with :

$$\mathbf{A} = \begin{pmatrix} 0.5 & -0.7 & 0.7 & 0.1 \\ 0.0 & 0.8 & 0.1 & 0.0 \\ 0.0 & 0.2 & 0.7 & 0.2 \\ -1.0 & 0.0 & 0.1 & 0.0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{C} = (0 \ 1 \ 0 \ 0)$$

According to the recurrences given in the previous section, we obtain the following results :

Initial step

$$\begin{aligned} \bar{\mathbf{A}}_{11}(0) &= 0.8 & \bar{\mathbf{B}}_1(0) &= 1 \\ \mathbf{R}(0) &= 1 & \mathbf{S}(0) &= 0 \\ \mathbf{P}(0) &= 0 & \mathbf{Q}(0) &= 1 \\ \mathbf{F}(0) &= z - 0.8 \end{aligned}$$

First step

$$\begin{aligned} \bar{\mathbf{A}}_{11}(1) &= 0.7 & \bar{\mathbf{B}}_1(1) &= (0.02 \ 0) \\ \mathbf{R}(1) &= z - 0.8 & \mathbf{S}(1) &= -1 \\ \mathbf{P}(1) &= \begin{pmatrix} \mathbf{R}(0) \\ \mathbf{P}(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \mathbf{Q}(1) &= \begin{pmatrix} \mathbf{S}(0) \\ \mathbf{Q}(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathbf{F}(1) &= z - 0.7 \end{aligned}$$

Second step

$$\begin{aligned} \bar{\mathbf{A}}_{11}(2) &= 0 & \bar{\mathbf{B}}_1(2) &= (0.02 \ 0 \ 0) \\ \mathbf{R}(2) &= z^2 - 1.5z + 0.54 & \mathbf{S}(2) &= -z + 0.7 \\ \mathbf{P}(2) &= \begin{pmatrix} \mathbf{R}(1) \\ \mathbf{P}(1) \end{pmatrix} = \begin{pmatrix} z-0.8 \\ 1 \\ 0 \end{pmatrix} & \mathbf{Q}(2) &= \begin{pmatrix} \mathbf{R}(1) \\ \mathbf{P}(1) \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \\ \mathbf{F}(2) &= z \end{aligned}$$

Third step

$$\begin{aligned}
 \bar{A}_{11}(3) &= 0.5 & \bar{B}_1(2) &= (5 \quad 7 \quad -0.7 \quad 0) \\
 R(3) &= z^3 - 1.5 z^2 + 0.52 z + 0.016 & S(3) &= -z^2 + 0.7 z + 0.02 \\
 P(3) &= \begin{pmatrix} R(2) \\ P(2) \end{pmatrix} = \begin{pmatrix} z^2 - 1.5 z + 0.54 \\ z - 0.8 \\ 1 \\ 0 \end{pmatrix} & Q(3) &= \begin{pmatrix} S(2) \\ Q(2) \end{pmatrix} = \begin{pmatrix} -z + 0.7 \\ -1 \\ 0 \\ 1 \end{pmatrix} \\
 F(3) &= z - 0.5
 \end{aligned}$$

The redundancy equation, obtained from (29) is then :

$$(z^4 - 2z^3 + 1.37 z^2 - 0.254 z - 0.08) y(k) = (z^3 - 1.2 z^2 + 0.43 z + 0.08) u(k)$$

EXAMPLE 2 (Massoumnia, 1988)

Let us consider the system in equations (3a) and (3b) with the definitions :

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Going through the details of the outlined procedure, it can be shown that for the first output $y_1(k)$ we obtain :

$$\begin{aligned}
 R(2) &= z(z - 2) & S(2) &= (2 - z \quad -z) \\
 C(3) &= 0 \\
 y_1(k+2) - 2 y_1(k+1) + 2 u_1(k) - u_1(k+1) - u_2(k+1) &= 0 & (37a)
 \end{aligned}$$

Repeating the procedure for the second output $y_2(k)$, we get :

$$\begin{aligned}
 R(1) &= z - 1 & S(1) &= (-1 \quad -1) \\
 C(2) &= (0 \quad 0) \\
 y_2(k+1) - y_2(k) - u_1(k) - u_2(k) &= 0 & (37b)
 \end{aligned}$$

With the assumption that the two actuators are fully reliable, the first redundancy equation is only sensitive to the first sensor and the second equation to the second sensor. These two equations enable the detection and localization of any sensor failure. The same equations have been found by Massoumnia (1988) by using the parity space approach (the reader should note the analogy between the minimal annihilating polynomial ψ used by Massoumnia and the polynomial R introduced here) ; however the latter technique needs the computation of the eigenvalues of the A matrix. Notice that equations (37a) and

(37b) allow other redundancy equations by linear combination to be generated ; in particular elimination of $u_1(k)$ or $u_2(k)$ between (37a) and (37b) gives two equations the first of which being sensitive to failure of the first actuator and the second being sensitive to the second actuator. This so-called aggregation procedure can be applied to locate gross errors on measured variables (Ragot, 1990).

CONCLUSION

The extraction of redundancy equations has been established for all dynamic linear state equations. They may be constructed for each measured output and give the so-called self-redundancy equations ; when it is desired to generate structured equations, these self-redundancy equations may be combined to form other equations sensitive to particular variables. The proposed algorithm uses a sequential elimination of the unmeasured variables ; the computation needs only the integration of discrete equations without matrix inversion or rank evaluation (if the different outputs are taken one after the other). With some modifications, the algorithm shown here can be applied to the generation of redundancy equations issued from unknown input systems and also from singular systems.

REFERENCES

- Barnett, V. and T. Lewis (1984). *Outliers in statistical data*. 2nd edition, Wiley and Sons, 1984.
- Basseville, M. and A. Benveniste (1986). *Detection of abrupt changes in signals and dynamical systems*. Lecture Notes in control and information sciences, 77, Springer Verlag.
- Bingulac, S.P. (1975). On the calculation of the transfer function matrix. *IEEE Trans. Aut. Control*, p. 134-135.
- Blackwell, C.C. (1984). On obtaining the coefficients of the output transfer function from a state space model and output model of a linear constant coefficient system. *IEEE Trans. Aut. Control*, AC-29 (12), p. 1122-1125.
- Chow, E.Y. and A.S. Willsky (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Aut. Control*, AC-29 (7), p. 603-615.
- Clark, R.N. (1978). A simplified instrument failure detection scheme. *IEEE Trans. Aero. and Electronic System*, AES-14 (4), p. 558-563.

Deckert, J.C., M.N. Desai, J.J. Deyst and A.S. Willsky (1977). F-8 DFBW sensor failure identification using analytical redundancy. *IEEE Trans. Aut. Control*, AC-22 (5), p. 795-803.

Ding, X. and P.M. Frank (1990). Fault detection via factorization approach. *Systems and Control Letters*, 14, p. 431-436.

Faddeev, D.K. and D.N. Faddeeva (1963). *Computational methods of linear algebra*. San Francisco, Freeman.

Fang, C.H. (1991). Determination of doubly coprime matrix fraction descriptions and corresponding Bezout identity solutions for a regular state-space system. *Systems and Control Letters*, 16, p. 45-48.

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

Frank, P.M. (1991). Fault diagnosis in dynamic systems using software redundancy. Lecture at the *2nd Makuhari Conference on High Technology Information Science and Computer Technology*, Chiba, Japan.

Gertler, J. and G.A. Almasy (1973). Balance calculations through dynamic system modelling. *Automatica*, 9 (1), p. 79-85.

Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation. *IFAC-IMACS proceedings of the Symposium "Safeprocess"*, Baden-Baden, vol. 1, p. 9-21.

Hashim, G., A. Rachid and C. Humbert (1988). Calculation of transfer function from state-model - a comparative study. *12th IMACS world congress on scientific computation*, Paris.

Iserman, R. (1984). Process fault detection based on modeling and estimation methods - a survey. *Automatica*, 20 (4), p. 387-404.

Kailath, T. (1980). *Linear systems*. Prentice Hall, Information and system sciences series, Englewood Cliffs, N.J.

Lou, X.C., A.S. Willsky and G.C. Verghese (1986). Optimally robust redundancy relations for failure detection in uncertain systems. *Automatica*, 22 (3), p. 333-344.

Magni, J.F. and Ph. Mouyon (1991). A generalized approach to observers for fault diagnosis. *30th IEEE Conf. Decision and Control*, Brighton.

Massoumnia, M.A. and W.E.V. Velde (1988). Generating parity relations for detecting and identifying control systems component failures. *J. Guidance*, 11 (1), p. 60-65.

Misra, P. and R.V. Patel (1987). Computation of transfer function matrices of linear multivariable systems. *Automatica*, 23 (5), p. 635-640.

Nett, C.N., C.A. Jacobson and M.J. Balas (1984). A connection between state-space and doubly coprime fractional representations. *IEEE Trans. Aut. Control*, AC-29 (9), p. 831-832.

Paraskevopoulos, P.N., M.A. Christodoulo and A.K. Boglu (1984). An algorithm for the computation of the transfer function matrix for singular systems. *Automatica*, 20, p. 259-260.

Patton, R.J., P.M. Frank and R.N. Clark (1989). *Fault diagnostic in dynamic systems : theory and application.*, Prentice Hall, 602 p.

Patton, R.J. and J. Chen (1991). A review of parity space approaches to fault diagnosis. *IFAC-IMACS proceedings of the Symposium "Safeprocess"*, Baden-Baden, vol. 1, p. 239-255.

Potter, J.E. and M.C. Suman (1977). Thresholdless redundancy management with arrays of skewed instruments. In *Integrity in electronic flight control systems*, AGARDograph (NATO), 224.

Ragot, J., M. Darouach, D. Maquin and G. Bloch (1990). *Validation de données et diagnostic*. Traité des nouvelles technologies, série diagnostic et maintenance, Hermès, 431 p.

Saidahmed, M.T.F. (1990). A new approach for designing a reduced-order controller of singular system. *IEEE Trans. Aut. Control*, AC-35 (4), p. 492-495.

Serth, R.W. and W.A. Heenan (1986). Gross error detection and data reconciliation in steam-metering systems. *AIChE Journal*, 32 (5), p. 733-742.

Staroswiecki, M., V. Cocquempot and J. Ph. Cassar (1991). Observer based and parity space approach for failure detection and identification. *IMACS symposium MCTS'91, Lille, France*.

Vaclavek, V. (1974). Gross systematic errors or biases in the balance calculation. *Papers of the Prague Inst. of Techn.*, Prague, Czechoslovakia.

Varga, A. and V. Sima (1981). Numerically stable algorithm for transfer function matrix evaluation. *International Journal of Control*, 33, p. 1123-1133.

Vidyasagar, M. (1985). *Control system synthesis : a factorization approach*. MIT Press, Cambridge, MA.

Viswanadham, N., J.H. Taylor and E.C. Luce (1987). A frequency-domain approach and isolation with application to GE-21 turbine engine control systems. *Control Theory and Advanced Technology*, 3 (1), p. 45-72.

Willsky, A.S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, 12 (6), p. 601-611.

Wünnenberg, J. (1989). Observer-based fault detection in dynamic systems. Ph. D. Thesis, University of Duisburg, p. 1-127.