# PCA-Based Fault Diagnosis in the Presence of Control and Dynamics

**Janos Gertler and Jin Cao**
School of Information Technology and Engineering, George Mason University, Fairfax, VA 22030

*In PCA-model-based fault diagnosis, using the isolation enhancement approaches of analytical redundancy, mis-isolation of some sensor and actuator faults may arise if the training data is collected under constant control. A detailed analysis is provided of how control actions affect PCA-model-based diagnosis. Ratio and feedback control in linear static and discrete dynamic systems, with full and partial PCA models is investigated. It is shown that all that it takes to eliminate the adverse effects is to vary the control set point (in feedback control) or the ratio coefficient (in ratio control) in the course of collecting the training data.* © 2004 American Institute of Chemical Engineers *AIChE J,* 50: 388–402, 2004
*Keywords: fault diagnosis, analytical redundancy, principal component analysis, ratio control, feedback control*

## Introduction

Principal Component Analysis (PCA) is a powerful tool in the monitoring of complex process systems. By analyzing the eigenstructure of the covariance matrix of data collected under normal operating conditions, linear relations among the variables are revealed; this allows the characterization of the high-dimensional process data in a representation space of significantly reduced dimension. A PCA model so created is then used as a basis of reference in process monitoring to indicate if the actual data somehow deviates from the expected "normal" process behavior. Fundamental results in the development of PCA as a process monitoring tool are due to MacGregor and coworkers (MacGregor and Kourti, 1995; Kourti and MacGregor, 1995, 1996), with additional important contributions by Wise and Ricker (1991), Piovoso and coworkers (Piovoso et al., 1992), Qin and coworkers (Dunia et al., 1996; Dunia and Qin, 1998), Cinar and coworkers (Negiz and Cinar, 1997; Raich and Cinar, 1997) and others.

An alternative approach to process monitoring, termed analytical redundancy (Willsky, 1976), relies on an explicit model of the plant, obtained from first principles or by systems identification. Under analytical redundancy, actual observations of the plant output are compared to values predicted by the model, using the input observations. Analytical redundancy may be implemented by a variety of techniques, including parity space (Chow and Willsky, 1984), consistency relations (Gertler and Singer, 1990; Gertler, 1998), and diagnostic observers (Frank, 1990). For a recent review of analytical redundancy methods, and the place in that context of the approach discussed in this article, the reader may turn to Gertler (2002).

While PCA-based monitoring has proven very effective in detecting abnormal process situations (fault detection), it has been found lacking when it came to pinpointing the root-cause of the problem (fault diagnosis). The standard approach to diagnosis has been the construction of contribution plots, comparing the relative deviation of each process variable from its predicted or nominal value (Kourti and MacGregor, 1996). Contribution plots do not identify the root of the problem, only narrow down the possible sources, leaving the interpretation to the process engineer.

Having compared PCA-based monitoring and analytical redundancy techniques, we came to the conclusion that there was a close relationship between the two approaches (Gertler and McAvoy, 1997). This suggested that it was possible to transplant the powerful fault isolation mechanisms of analytical redundancy, particularly the notion of structured residuals, into the PCA framework. We proposed two techniques. One uses partial PCA models, which cover subsets of variables and are selectively sensitive to subsets of faults (Gertler and McAvoy, 1997). The other method relies on the full PCA model and uses algebraic projection to achieve selective fault sensitivity (Gertler et al., 1999). Both methods utilize the residual space, the orthogonal complement of

Correspondence concerning this article should be addressed to J. Gertler at jgertler@gmu.edu.

the representation space of the PCA model. And both methods require that certain minimal persistent excitation conditions be met during the model-building phase. In the articles quoted above, the new methods were successfully applied to a subsystem (the reactor) of the Tennessee Eastman benchmark process.

Recently, Yoon and MacGregor (2000a) published an article in which they analyze the "relationship between statistical and causal model based approaches." In their view, the persistent excitation requirement is the fundamental issue; while traditional PCA relies entirely on "normal operating data," causal models require "designed experiments" (or first principle knowledge). According to this interpretation, PCA-based methods which do require persistent excitation actually belong to the causal model based class. In a subsequent article (Yoon and MacGregor, 2000b), they propose a fault isolation method which utilizes the directional properties of the fault effects in the representation and the residual space.

In this article, we wish to revisit the issue of persistent excitation in PCA-based diagnosis. More specifically, we wish to investigate the effect of control during the modeling and the monitoring phases. In doing so, we will rely on the experience gained when applying our ideas to the Tennessee Eastman simulator. We will consider ratio control and feedback control and compare them to the open-loop case. We will explore the behavior of fault-responses in the various situations and its effect on fault isolation by algebraic projection and by partial PCA models. We will show that, in the modeling phase, some variation in the control ratio (ratio control) or in the set point (feedback control) is all that is needed, as far as the controllers are concerned, to satisfy persistent excitation. Also, once a model has been so obtained, in the monitoring phase it is irrelevant whether the system operates with or without control. The directional properties of the fault-responses will be an important element (and a natural byproduct) of the analysis.

Our investigations here will be concerned with sensor and actuator faults in linear systems. We will first discuss static systems; then, we will show how our techniques and results can be extended and applied to discrete dynamic models. The isolation of plant faults requires an additional modeling effort and is outside the scope of the present study. Also, we will not consider unmeasured disturbances in this work. The fault isolation properties of the various schemes will be illustrated on artificial examples (rather than more extensive plant models) to keep a focus on the points to demonstrate.

Some fundamental concepts are defined. Standard PCA modeling is reviewed. The generation of isolation enhanced residuals from the static PCA model and the effect of control constraints on such residuals is discussed. The static design methods are extended to discrete dynamic systems. The effect of control is analyzed in the dynamic environment.

## Fundamentals

In this section, we define a few fundamental concepts concerning variables, equations and faults.

### *Variables*

We consider a vector of $n$ process variables

$$\mathbf{x}(t) = [x_1(t) \cdots x_n(t)]' \tag{1}$$

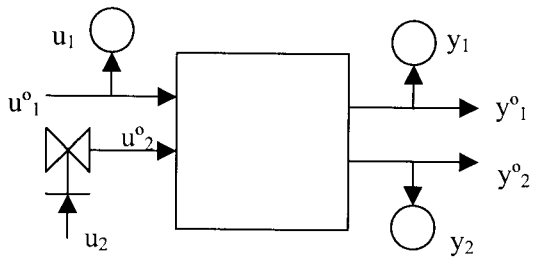


**Figure 1. Observations and true variable values.**

Here $'$ means transpose, while $t$ is the discrete time variable. In a static framework, the actual value of time is not relevant, so $t$ may be considered simply as a sequential index for consecutive samples (observations) of $\boldsymbol{x}$.

When dealing with process units, it is usually meaningful to distinguish input and output variables, the inputs $\boldsymbol{u}(t)$ being physical causes while the outputs $\boldsymbol{y}(t)$ the physical effects. Thus, with $k$ inputs and $m$ *outputs*

$$\mathbf{u}(t) = [u_1(t) \cdots u_k(t)]' \quad \mathbf{y}(t) = [y_1(t) \cdots y_m(t)]' \tag{2}$$

and $\boldsymbol{x}(t)$ is the combined vector of $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$, so that $n = k + m$ and

$$\mathbf{x}(t) = [\mathbf{u}'(t) \quad \mathbf{y}'(t)]' \tag{3}$$

We call the *observation* of a variable its value that is available for modeling and monitoring; this is the measurement value for a measured variable and the command value for a manipulated variable. Since all computations are performed on the observations, we keep the basic symbols for those values. In contrast, we denote the *true* values of the variables, that act on and arise from the physical plant (and are used in derivations), as $\boldsymbol{u}^0(t)$, $\boldsymbol{y}^0(t)$ and $\boldsymbol{x}^0(t)$. For a simple two-input two-output system, which we will refer to repeatedly in the course of the article, the situation is depicted in Figure 1.

### *Equations*

By equations, we mean the linear relationships that exist among the variables. We will distinguish two types of equations.

(1) *Plant equations.* These describe the internal behavior of plant units, that is, how the outputs of the unit depend on its inputs. Plant equations always link *true variables.*

(2) *Control equations.* These describe the constraints placed on the variables by control actions. Such actions may be considered to be outside the plant units. Control equations always link command values to each other or to measurement values, that is, they concern *observed variables.*

While it may be somewhat arbitrary what we consider occurring inside or outside a plant unit (or even what a plant unit is), the type of variables (true or observed) always clearly identifies the type of the equation.

*Plant Equations.* The general form of static plant equations is

$$\mathbf{B}\mathbf{x}^0(t) = \mathbf{0} \tag{4}$$

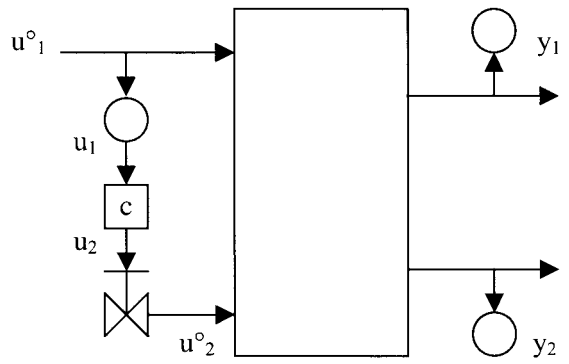


**Figure 2. Ratio control**

When input and output variables can be defined separately, the input-output relationship is

$$\mathbf{y}^0(t) = \mathbf{A}\mathbf{u}^0(t) \tag{5}$$

With Eq. 3 and 4, this implies

$$\mathbf{B} = [\mathbf{A} \quad -\mathbf{I}] \tag{6}$$

where $\boldsymbol{I}$ is an $m \cdot m$ identity matrix. Thus, Eq. 4 represents $m$ linear relationships among $k + m$ true variables.

Note that sometimes plant inputs are linearly related, due to some control action "upstreams." Locally, these are measured variables and the linear relation concerns their true value. Thus, the relation is a plant equation. A special case of this is an input variable whose true value is constant.

*Control Equations.* The general form of static control equations is

$$\mathbf{C}\mathbf{x}(t) = \mathbf{0} \tag{7}$$

where $\boldsymbol{C}$ is a $\mu \cdot (k + m)$ matrix, so that Eq. 7 represents $\mu$ linear relations among $k + m$ observed variables. We consider two types of control actions: ratio control and feedback control.

(a) *Ratio control.* Under ratio control, two inputs $u_i(t)$ and $u_j(t)$ are related to each other as

$$u_j(t) = cu_i(t) \tag{8}$$

There are two possible scenarios:

(i) $u_i(t)$ is measured while $u_j(t)$ is manipulated, in which case Eq. 8 contains the measured value for the former and the command value for the latter (Figure 2);

(ii) Both variables are manipulated (two values determined by some control algorithm), in which case they both appear in Eq. 8 with their command value.

(b) *Feedback control.* Under feedback control, an input $u_j(t)$ is related to an output $y_i(t)$. We consider proportional and integrating feedback.

(i) The proportional feedback relation is

$$u_j(t) = K_i\,[s_i(t) - y_i(t)] \tag{9}$$

Here, $s_i$ is the set point for $y_i(t)$. In the equation, $y_i(t)$ appears with its measured value while $u_j(t)$ with its command value (Figure 3).

(ii) An integrating controller, in steady state, is characterized as

$$y_i(t) = s_i(t) \tag{10}$$

Now only $y_i(t)$ is subject to a control equation, which applies to its measured value. The steady-state effect of integration is identical with the proportional controller (Eq. 9) as $K_i \to \infty$. (The value of $u_j^0(t)$ can be computed backward from the plant equations.)

As we will show below, the presence of control constraints (equations) in the data from which the model is built creates some difficulties in fault isolation. However, for the purpose of modeling (PCA or other), each of the above control algorithms constitutes a control constraint only if it represents the *same* linear relation over the entire training dataset. The ratio-control constraint (equation) can be "removed" from the modeling if the training data is collected with the coefficient $c$ varying (taking at least two different values). Similarly, the feedback control constraint (proportional or integrating) can be removed if the training data is collected with the set point varying (taking at least two different values). Such variations may be injected intentionally, or simply the training data may be taken from such operating conditions when they occur naturally.

### *Faults*

In this article, we are dealing only with sensor and actuator faults. We represent these as unknown quantities that enter the relationship between the observed and true values of the concerned variable in an additive way. For a measured input or output, the relationship is

$$u_i(t) = u_i^0(t) + \Delta u_i(t) \quad y_i(t) = y_i^0(t) + \Delta y_i(t) \tag{11}$$

where $\Delta u_i(t)$ and $\Delta y_i(t)$ are the sensor faults. For a manipulated input, we have

$$u_i^0(t) = u_i(t) + \Delta u_i(t) \quad \text{or} \quad u_i(t) = u_i^0(t) - \Delta u_i(t) \tag{12}$$

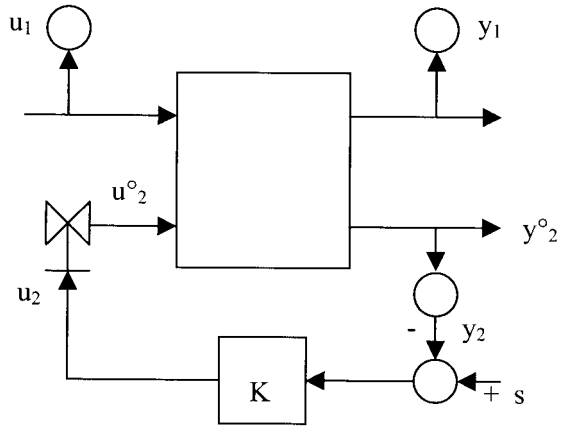


**Figure 3. Feedback control**

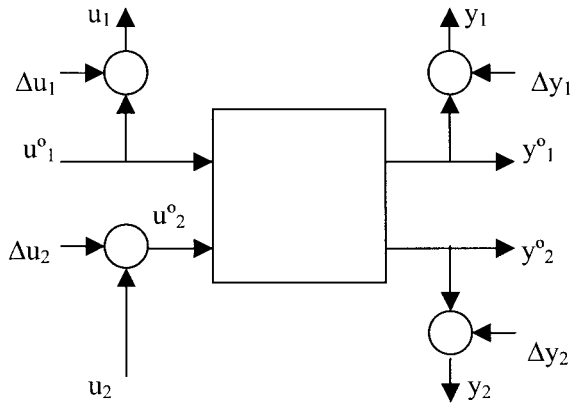


**Figure 4. True variables, faults and observations**

where $\Delta u_i(t)$ is now the actuator fault. For our two-input two-output system, the relationship between observations, true variables and faults is shown in Figure 4.

The relations (Eqs. 11 and 12) may be combined into a single equation for all variables, as

$$\mathbf{x}(t) = \mathbf{x}^0(t) + \Delta\mathbf{x}(t) \tag{13}$$

where

$$\Delta x_i(t) = \begin{bmatrix} \Delta u_i(t) \text{ for an input sensor} \\ -\Delta u_i(t) \text{ for an actuator} \\ \Delta y_i(t) \text{ for an output sensor} \end{bmatrix} \tag{14}$$

Yoon and MacGregor (2000b) distinguish two fault categories: simple and complex faults. A simple fault is one that only affects the (observed) variable it is associated with. Such is the fault of a sensor that is not used in a control loop. A complex fault, on the other hand, propagates to other process variables. These are faults of sensors and actuators that participate in control schemes. Strictly speaking, an actuator fault is complex even if no control is involved since its effect is propagating to one or more outputs. For the purpose of the present study, we need to tighten this definition. We will speak of an

- *In-loop fault* if the variable with which the faulty instrument is associated does appear in a control equation; and
- *Off-loop fault* if the concerned variable does not appear in any control equation. In this sense, the fault of an actuator operating in open loop is an off-loop fault.

When the plant equations are applied to the observed variables, they provide information about the faults. With Eqs. 4 and 13

$$\mathbf{B}\mathbf{x}(t) = \mathbf{B}[\mathbf{x}^0(t) + \Delta\mathbf{x}(t)] = \mathbf{B}\Delta\mathbf{x}(t) \tag{15}$$

However, the control equations $\mathbf{C}\mathbf{x}(t) = \mathbf{0}$ carry no information concerning the faults. This has fundamental implications which will be explored later in the article.

## Standard PCA Modeling

The standard PCA model is constructed by the eigenstructure analysis of the covariance matrix of normal operating data. Consider a set of "training" data $\boldsymbol{x}(\tau)$, $\tau = 1 \cdots N$. It is assumed that the training data is free from faults, that is, $\boldsymbol{x}(\tau) = \boldsymbol{x}^0(\tau)$ for all $\tau$. Construct the covariance matrix

$$\Phi = \frac{1}{N}\sum_{\tau=1}^{N} \mathbf{x}(\tau)\mathbf{x}'(\tau) \tag{16}$$

and obtain the eigenvalues $\sigma_j^2$ and eigenvectors $\boldsymbol{q}_j$ of the covariance matrix, where $j = 1 \cdots n$. Then any element $\boldsymbol{x}(\tau)$ of the training set may be described as

$$\mathbf{x}(\tau) = \sum_{j=1}^{n} \mathbf{q}_j z_j(\tau) \tag{17}$$

where

$$z_j(\tau) = \mathbf{q}_j'\mathbf{x}(\tau) \tag{18}$$

is the projection of $\boldsymbol{x}(\tau)$ on $\boldsymbol{q}_j$. If there are linear relations among the elements of the $\boldsymbol{x}(\tau)$ vector, those reduce the rank of the covariance matrix and make some of the eigenvalues zero. Then, $\boldsymbol{x}(\tau)$ can be reproduced as

$$\mathbf{x}(\tau) = \sum_{j=1}^{n'} \mathbf{q}_j z_j(\tau) \quad n' < n \tag{19}$$

If there are $k$ inputs and $m$ outputs, and only plant relations exist, then $n' = k$. If, in addition, there are $\mu$ control equations then $n' = k - \mu$.

Equation 19 can also be written, in more compact form, as

$$\mathbf{x}(\tau) = \mathbf{Q}_M\mathbf{z}(\tau) \quad \mathbf{z}(\tau) = \mathbf{Q}_M'\mathbf{x}(\tau) \tag{20}$$

where

$$\mathbf{Q}_M = [\mathbf{q}_1 \cdots \mathbf{q}_{n'}] \quad \mathbf{z}(\tau) = [z_1(\tau) \cdots z_{n'}(\tau)]' \tag{21}$$

The first $n'$ eigenvectors, $\boldsymbol{q}_1 \cdots \boldsymbol{q}_{n'}$, span the representation or model space, while the remaining eigenvectors, $\boldsymbol{q}_{n'+1} \cdots \boldsymbol{q}_n$, span the residual space. Note that, ideally (with no noise and nonlinearities), the data does not fully define the $\boldsymbol{q}_{n'+1} \cdots \boldsymbol{q}_n$ eigenvectors, only the space they span; the actual directions depend on secondary conditions (noise, and so on) and the heuristics built into the eigenstructure algorithm.

## Isolation Enhanced Residuals

In analytical redundancy based diagnosis, residuals are generated that express the discrepancy between the predicted and actual process situation. These residuals are then usually enhanced to support fault isolation. There are two major enhancement techniques:

- *Directional Residuals,* characterized by fault-specific response directions in the residual space (Jones, 1973);
- *Structured Residuals,* characterized by selective decoupling from subsets of faults, resulting in fault-specific Boolean patterns (Gertler and Singer, 1990).

In a static framework, directionality is an automatic property of residuals. Ideally, the response directions should be orthogonal; this is not automatic, but can be achieved by a transformation, provided the number of faults does not exceed that of the original "primary" residuals (that is, the number of the outputs).

Structured residuals need to be designed. In a geometric sense, structured behavior may be achieved by projecting the primary residuals onto a subspace orthogonal to selected response directions. The best design aims at a one-dimensional (1-D) subspace, because this way the residual may be decoupled from the maximum number of faults, and because 1-D residuals are the easiest to evaluate. Each residual can be decoupled from as many faults as the number of outputs minus one.

We will show below how directional and structured residuals arise from the standard PCA model. We will also review the "partial PCA" technique of generating structured residuals. In either case, we will point out the difficulties stemming from the presence of control equations in the model.

Note that, while traditional PC analysis is usually applied to the entirety of large systems (and that is where its real strength lies), enhanced residuals would more typically be generated for subsystems. Analytical redundancy considers the consistency of a group of variables, with each such group containing the complete set that intersects a closed contour around any part of the system. The complexity of any set may be limited, while there may be a large number of sets, with multiple partial overlaps.

### *Enhanced residuals from the standard PCA model*

In the monitoring phase of PCA-based diagnosis, the actual observations $\boldsymbol{x}(t)$ are projected both into the model space and into the residual space. If there is no fault, the projection into the latter (the residual) is ideally zero (in reality, it is not, due to noise, disturbances, modeling errors). In response to a fault, there is an increment in the model space and a residual in the residual space. Both of these carry information concerning the fault and may contribute to the diagnosis (Yoon and MacGregor, 2000b). The increment, however, may be difficult to utilize, because it appears on top of the normal variations of the process. This is especially troublesome if the fault itself is also a slow drift. Therefore, we prefer to build the diagnosis entirely on the residual.

When the PCA model (Eq. 20) is applied to an observation $\boldsymbol{x}(t)$ outside the training set, it usually returns a nonzero residual. This is computed as

$$\boldsymbol{\epsilon}(t) = \mathbf{x}(t) - \mathbf{Q}_M\mathbf{Q}_M'\mathbf{x}(t) \tag{22}$$

where the second term is the projection of $\boldsymbol{x}(t)$ into the model space. Due to the orthonormality of the eigenvectors, Eq. 22 can also be written as

$$\boldsymbol{\epsilon}(t) = \mathbf{Q}_R\mathbf{Q}_R'\mathbf{x}(t) \tag{23a}$$

where

$$\mathbf{Q}_R = [\mathbf{q}_{n'+1} \cdots \mathbf{q}_n] \tag{23b}$$

$\boldsymbol{\epsilon}(t)$ in Eq. 23a is clearly the projection of $\boldsymbol{x}(t)$ onto the residual space; it is written in terms of the original $x$ coordinates. The same in terms of the $\mathbf{q}_{n'+1} \cdots \mathbf{q}_n$ coordinates is

$$\mathbf{e}(t) = \mathbf{Q}_R'\mathbf{x}(t) \tag{24}$$

The vector $\boldsymbol{e}(t)$ is the PCA residual and Eq. 24 is the formula to compute it from the observation $\boldsymbol{x}(t)$.

Assume first that the system model is obtained solely from plant equations, with no control equations present. Then with the training data (without any noise and modeling error), $\boldsymbol{Q}_R'\boldsymbol{x}(\tau) = \mathbf{0}$. Recall that $\boldsymbol{B}\boldsymbol{x}(\tau) = \mathbf{0}$. Thus, the rows of both $\boldsymbol{Q}_R'$ and $\boldsymbol{B}$ are orthogonal to the representation space and they both span the residual space. This implies

$$\mathbf{Q}_R' = \mathbf{MB} \tag{25}$$

where $\boldsymbol{M}$ is some full-rank square matrix. Then, Eq. 24 may be written as

$$\mathbf{e}(t) = \mathbf{MBx}(t) \tag{26}$$

Finally, with Eq. 15, this becomes

$$\mathbf{e}(t) = \mathbf{MB}\Delta\mathbf{x}(t) = \mathbf{Q}_R'\Delta\mathbf{x}(t) \tag{27}$$

Note that this is valid even if control equations are then present in the monitoring data. Denoting the columns of $\boldsymbol{Q}_R'$ as $\boldsymbol{q}_{*1} \cdots \boldsymbol{q}_{*n}$, Eq. 27 may be expanded as

$$\mathbf{e}(t) = [\mathbf{q}_{*1} \cdots \mathbf{q}_{*n}][\Delta x_1(t) \cdots \Delta x_n(t)]' \tag{28}$$

That is, the residual response to each fault is the respective column of the $\boldsymbol{Q}_R'$ matrix. Note that the vectors $\boldsymbol{q}_{*1} \cdots \boldsymbol{q}_{*n}$ are not orthonormal. Equation 28 provides an easy way to find the response directions without the need for active experimentation or historical records involving particular faults. These response directions may then be used in an angle-based scheme (Yoon and MacGregor, 2000b) or any other way. For easier isolation, the directions for a selected subset of $m$ faults may be made orthogonal by the transformation

$$\mathbf{r}(t) = \mathbf{Ve}(t) \tag{29}$$

with $\boldsymbol{V}$ so chosen that $VQ_R^* = \boldsymbol{I}$, where $\boldsymbol{Q}_R^*$ is a square sub-matrix of $\boldsymbol{Q}_R'$. Further, structured residuals may be obtained by the transformation (Eq. 29) by choosing $\boldsymbol{V}$ so that its rows are orthogonal to selected $\boldsymbol{q}_{*j}$ columns, in the desired pattern.

*The Effect of Control Equations in the Model.* Now consider the case when control equations are also present in the system, and they are identical both in the training data and in the testing data. Control equations in the training data increase the dimension of the residual space. Also, now the rows of the combined $\boldsymbol{B}$ and $\boldsymbol{C}$ matrix span the same space as the rows of $\boldsymbol{Q}_R'$, so that

$$\mathbf{Q}_R' = [\mathbf{M}_B\ \mathbf{M}_C]\begin{bmatrix}\mathbf{B}\\ \mathbf{C}\end{bmatrix} \tag{30}$$

where $[\boldsymbol{M}_B\ \boldsymbol{M}_C]$ is a square, invertible matrix. The columns of $\boldsymbol{Q}_R'$ are now

$$\mathbf{q}_{*j} = \mathbf{M}_B\mathbf{b}_{\cdot j} + \mathbf{M}_C\mathbf{c}_{\cdot j} \tag{31}$$

Expand the control equations as $\boldsymbol{Cx}(t) = [\boldsymbol{c}_1 \cdots \boldsymbol{c}_n]\ [\boldsymbol{x}_1(t) \cdots \boldsymbol{x}_n(t)]'$. If a variable $\boldsymbol{x}_j(t)$ does not appear in the control equation, its coefficient is $\boldsymbol{c}_j = \mathbf{0}$. Now the residual $\boldsymbol{e}(t)$ is

$$\mathbf{e}(t) = \mathbf{Q}_R'\mathbf{x}(t) = (\mathbf{M}_B\mathbf{B} + \mathbf{M}_C\mathbf{C})\mathbf{x}(t) = \mathbf{M}_B\mathbf{B}\mathbf{x}(t) + \mathbf{M}_C\mathbf{C}\mathbf{x}(t) = \mathbf{M}_B\mathbf{B}\Delta\mathbf{x}(t) \tag{32}$$

In response to a fault $\Delta\boldsymbol{x}_j(t)$, the residual becomes

$$\mathbf{e}(t|\Delta x_j) = \mathbf{M}_B\mathbf{b}_{\cdot j}\Delta\mathbf{x}_j(t) \tag{33}$$

Comparing Eqs. 31 and 33 reveals that, for off-loop faults whose related variable does not appear in the control equations ($\boldsymbol{c}_j = \mathbf{0}$), the respective columns of $\boldsymbol{Q}_R'$ are still the fault-response directions. However, for in-loop faults whose related variable does appear in the control equations ($\boldsymbol{c}_j \neq \mathbf{0}$), this is no longer the case. This is because the $\boldsymbol{C}$ matrix, while affecting $\boldsymbol{Q}_R'$, now does not contribute to the fault responses. Thus, for directional fault isolation or for structured design, those response directions need to be found by a special experiment or from historical data. Alternatively, the training data needs to be collected with the linear control constraints removed.

Note that if the training data was collected under control constraints, but those constraints are not present in the testing data, then the model is fundamentally violated, resulting in (large) nonzero residuals even if there are no faults.

Note also that, if plant inputs are linearly related due to some "upstream" action and measured locally, then the local sensors are off-loop and the linear relation does not affect the isolation of faults acting on them. However, if the linear relation is removed in the monitoring phase, then this results in a violation of the model and leads to nonzero residuals.

### *Illustrative example*

To keep the concepts in focus, we will consider the simple two-input two-output system depicted in Figure 1. It should be emphasized that real systems, even after decomposition, are (significantly) larger. This implies more freedom in the design of the residuals, but also increased complexity of the computations.

The plant equations for our example system are

$$y_1^0(t) = 2u_1^0(t) + u_2^0(t)$$
$$y_2^0(t) = 4u_1^0(t) + 3u_2^0(t)$$

The two inputs were driven by white Gaussian random sequences, with unit variance.

(1) *No control relations in the training data.* A PCA model is obtained; it is 2-D, with a 2-D residual space. The two eigenvectors of the residual space are given below. The slopes of the columns (the predicted fault responses) are also shown.

| | | | | |
|---|---|---|---|---|
| $\mathbf{q}_3'$ | 0.8230 | 0.5254 | −0.1836 | −0.1139 |
| $\mathbf{q}_4'$ | −0.0920 | 0.3520 | 0.8421 | −0.3981 |
| $q_{4j}/q_{3j}$ | −0.1118 | 0.6700 | −4.5866 | 3.4951 |

Then, test data is generated with faults injected, one at a time, still without any control. The observations are projected onto the residual subspace and the directions of the residuals are obtained as

| response to | $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|---|
| $e_4/e_3$ | −0.1118 | 0.6705 | −4.5866 | 3.4952 |

Clearly, the response directions agree with the ones predicted from $\boldsymbol{Q}_R'$. The experiment is then repeated, still with no control in the training data, but this time with test data collected under ratio control and under feedback control. The residual responses are identical with the above.

(2) *Training and testing under ratio control* (Figure 2). The training data is generated under the control constraint $u_2(\tau) = cu_1(\tau)$, with $c = 1.5$. The residual space is now 3-D. The slopes of the columns of $\boldsymbol{Q}_R'$ (in the two planes) are obtained as

| | | | | |
|---|---|---|---|---|
| $q_{3j}/q_{2j}$ | −2.0479 | −4.8293 | 0.3131 | −0.1947 |
| $q_{4j}/q_{2j}$ | 7.2403 | −1.5988 | −0.0973 | 0.0548 |

Testing data is then generated, under the same ratio control, with one fault at a time. The residual directions are

| response to | $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|---|
| $e_3/e_2$ | 7.5855 | −1.4574 | 0.3131 | −0.1947 |
| $e_4/e_2$ | −2.2747 | 0.4328 | −0.0973 | 0.0548 |

Clearly, the responses to $\Delta y_1$ and $\Delta y_2$ (off-loop faults) are as predicted from $\boldsymbol{Q}_R'$ but those to $\Delta u_1$ and $\Delta u_2$ (in-loop faults) are different.

(3) *Training and testing under feedback control* (Figure 3). The training data is generated under the control constraint $u_2(\tau) = K[s - y_2(\tau)]$, with $s = 0$ and $K = 1$. The residual space is now 3-D. The slopes of the columns of $\boldsymbol{Q}_R'$ (in the two planes) are obtained as

| | | | | |
|---|---|---|---|---|
| $q_{3j}/q_{2j}$ | −0.8436 | 11.6940 | 11.6940 | 0.5862 |
| $q_{4j}/q_{2j}$ | 0.0000 | −16.5982 | 16.5982 | 0.0000 |

Testing data is then generated, under the same feedback control, with one fault at a time. The residuals are

| response to | $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|---|
| $e_3/e_2$ | −0.8436 | −0.3278 | 11.6940 | 0.5862 |
| $e_4/e_2$ | 0.0000 | 0.6829 | 16.5982 | 1.8928 |

Now, the responses to $\Delta u_1$ and $\Delta y_1$ (off-loop faults) are as predicted from $\boldsymbol{Q}_R'$, while those to $\Delta u_2$ and $\Delta y_2$ (in-loop faults) are different.

(4) *Training under Ratio Control, Varying Control Gain.*

The training data is now collected under ratio control, but the control gain is $c = 1.5$ for half of the data and $c = 2.5$ for the other half. This control imposes no linear relation between the two inputs. The PCA model now yields a 2-D model space and a 2-D residual space. The latter is obtained as

| | | | | |
|---|---|---|---|---|
| $\mathbf{q}_3'$ | 0.6216 | 0.6323 | 0.3322 | −0.3215 |
| $\mathbf{q}_4'$ | −0.5471 | −0.0126 | 0.7953 | −0.2609 |
| $q_{4j}/q_{3j}$ | −0.8800 | −0.0200 | 2.3941 | 0.8114 |

The eigenvectors $\mathbf{q}_3'$ and $\mathbf{q}_4'$ are not the same as the ones obtained from data with no control, but they do not need to. They do need, however, to span the same residual space which they actually do; the above two eigenvectors can be obtained from the ones computed in the no-control case by the transformation $\boldsymbol{Q}'_{R,\text{ ratio-control}} = \boldsymbol{M}\boldsymbol{Q}'_{R,\text{ no-control}}$, where (as computed from the data)

$$\mathbf{M} = \begin{bmatrix} 0.8198 & 0.5734 \\ -0.5734 & 0.8198 \end{bmatrix}$$

The fault-response directions obtained from testing data, generated with or without ratio control, agree with the predicted directions. Thus, the model generated under ratio control with varying control gain (just two values) behaves like the model generated with no control present.

(5) *Training under feedback control, varying set point.* The training data is now collected under feedback control, with $y_2(\tau)$ measured and $u_2(\tau)$ manipulated, with the set point being $s = -1$ for half of the data and $s = 1$ for the other half. This imposes no linear control relation on $y_2(\tau)$ and $u_2(\tau)$. Both the model space and the residual space are 2-D. The slopes of the latter are obtained as

| | | | | |
|---|---|---|---|---|
| $q_{4j}/q_{3j}$ | −0.0312 | 0.7931 | −3.2929 | 4.9685 |

The fault-response directions obtained from testing data, generated with or without feedback control, agree with the predicted directions. Thus, the model generated under feedback control with varying set point (just two values) behaves like the model generated with no control present.

(6) *Training and testing with linearly dependent inputs.* In this demonstration, the true values of the input variables, both measured, are linearly related, $u_2^0(\tau) = cu_1^0(\tau)$, with $c = 1.5$. Thus, there are three linear relations in the data but none of them concerns the observed values. The 3-D residual space is exactly the same as the one obtained under ratio control, constant gain. The fault-response directions, obtained experimentally, are

| response to | $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|---|
| $e_3/e_2$ | −2.0479 | −4.8293 | 0.3131 | −0.1947 |
| $e_4/e_2$ | 7.2403 | −1.5988 | −0.0973 | 0.0548 |

Unlike in the ratio-control case, these response directions are now in perfect agreement with the slopes predicted from the eigenvectors. Thus, linear dependence concerning the true value of the inputs increases the dimension of the residual space but does not create any difficulty in fault isolation.

### *Structured residuals from partial PCA models*

Structured residuals may be generated by PCA without any algebraic transformation, using the idea of partial PCA models (Gertler and McAvoy, 1997). Partial PCA models, obtained directly from training data, describe the relationships among subsets of variables, according to a specific Boolean structure. Each subset of variables is so chosen that there is exactly one linear relation among them; we will refer to this relation as the subsystem model. Clearly, this subsystem model is violated if a sensor or actuator associated with any of the variables in the subset is faulty, but is insensitive to faults associated with variables outside the subset. Thus, selective sensitivity to subsets of faults, that is the essence of structured residuals, is achieved.

Consider first the case when there are no control relations in the training data. Then the system is described as $\boldsymbol{B}\boldsymbol{x}^0(\tau) = \boldsymbol{0}$. This set contains $m$ equations and $m + k$ variables. Apply a transformation as

$$\mathbf{v}_i'\mathbf{B}\mathbf{x}^0(\tau) = 0 \tag{34}$$

where $\boldsymbol{v}_i'$ is a row vector. This is a single equation. Decompose $\boldsymbol{B}$ as $\boldsymbol{B} = [\boldsymbol{B}^i \ \boldsymbol{B}^{i\#}]$, where $\boldsymbol{B}^{i\#}$ contains $m-1$ selected columns of $\boldsymbol{B}$. The vector $\boldsymbol{v}_i'$ can be so chosen that

$$\mathbf{v}_i'\mathbf{B}^{i\#} = \mathbf{0} \tag{35}$$

Then Eq. 34 becomes

$$\mathbf{w}_i'\mathbf{x}_i^0(\tau) = 0 \tag{36}$$

where

$$\mathbf{w}_i' = \mathbf{v}_i'\mathbf{B}^i \tag{37}$$

and $\boldsymbol{x}_i^0(\tau)$ is a subset of $\boldsymbol{x}^0(\tau)$, containing $k + 1$ elements. This procedure can also be thought of as using $m - 1$ of the original equations to eliminate $m - 1$ variables. The system described by Eq. 36 can be modeled by PCA; this is a single linear relation among $k + 1$ variables, resulting in a $k$ dimensional model space and a 1-D residual space. This is a partial PCA model or subsystem model. The sole eigenvector of the residual space is co-linear with $\boldsymbol{w}_i'$; if the latter is normalized then

$$\mathbf{Q}_{Ri}' = \mathbf{w}_i' \tag{38}$$

Now, if Eq. 36 is applied to observed data $\boldsymbol{x}(t)$ then a residual arises which is

$$r_i(t) = \mathbf{w}_i'\mathbf{x}_i(t) = \mathbf{v}_i'\mathbf{B}^i\mathbf{x}_i(t) = \mathbf{v}_i'\mathbf{B}\mathbf{x}(t) = \mathbf{v}_i'\mathbf{B}\Delta\mathbf{x}(t) = \mathbf{w}_i'\Delta\mathbf{x}_i(t) \tag{39}$$

That is, the residual responds to the faults $\Delta\boldsymbol{x}_i(t)$, but is completely decoupled from all the other faults. Thus, it is a structured residual. Creating several partial PCA models, each in a different structure following a structure matrix, leads to a structured fault isolation scheme.

*The effect of control equations.* If control relations $\boldsymbol{C}\boldsymbol{x}(\tau) = \boldsymbol{0}$ are present when the training data is collected, then the full set of equations is

$$\begin{bmatrix}\mathbf{B}\\ \mathbf{C}\end{bmatrix}\mathbf{x}(\tau)=\mathbf{0} \tag{40}$$

With $\mu$ control relations, $m+\mu-1$ variables may be eliminated, so a partial model containing a single equation covers $k-\mu+1$ variables. The transformation $\mathbf{v}_i'$ is now so chosen that

$$\mathbf{v}_i'\begin{bmatrix}\mathbf{B}^{i\#}\\ \mathbf{C}^{i\#}\end{bmatrix}=[\mathbf{v}_{Bi}'\ \mathbf{v}_{Ci}']\begin{bmatrix}\mathbf{B}^{i\#}\\ \mathbf{C}^{i\#}\end{bmatrix}=\mathbf{0} \tag{41}$$

Recall that not all elements of $\mathbf{x}(\tau)$ are necessarily included in the control relations. It follows from Eq. 41 that

$$\begin{aligned}&\mathbf{v}_{Bi}'\mathbf{b}_{\cdot j}=0 \quad \text{if} \quad \mathbf{c}_{\cdot j}=\mathbf{0} \quad \text{(not included in control)}\\ &\mathbf{v}_{Bi}'\mathbf{b}_{\cdot j}\neq 0 \quad \text{if} \quad \mathbf{c}_{\cdot j}\neq\mathbf{0} \quad \text{(included in control)}\end{aligned} \tag{42}$$

(where $\mathbf{b}_{\cdot j}$ and $\mathbf{c}_{\cdot j}$ are columns of $\boldsymbol{B}^{i\#}$ and $\boldsymbol{C}^{i\#}$). Now the residual is

$$\begin{aligned}r_i(t)&=\mathbf{w}_i'\mathbf{x}_i(t)=\mathbf{v}_i'\begin{bmatrix}\mathbf{B}^{i}\\ \mathbf{C}^{i}\end{bmatrix}\mathbf{x}_i(t)=\mathbf{v}_i'\begin{bmatrix}\mathbf{B}^{i}\\ \mathbf{C}^{i}\end{bmatrix}\mathbf{x}_i(t)+\mathbf{v}_i'\begin{bmatrix}\mathbf{B}^{i\#}\\ \mathbf{C}^{i\#}\end{bmatrix}\mathbf{x}_{i\#}(t)\\ &=\mathbf{v}_i'\begin{bmatrix}\mathbf{B}\mathbf{x}(t)\\ \mathbf{C}\mathbf{x}(t)\end{bmatrix}=\mathbf{v}_i'\begin{bmatrix}\mathbf{B}\Delta\mathbf{x}(t)\\ \mathbf{0}\end{bmatrix}=\mathbf{v}_{Bi}'\mathbf{B}^{i}\Delta\mathbf{x}_i(t)+\mathbf{v}_{Bi}'\mathbf{B}^{i\#}\Delta\mathbf{x}_{i\#}(t)\end{aligned} \tag{43}$$

Here, $\mathbf{x}_{i\#}(t)$ is the subset of variables eliminated from the model and $\Delta\mathbf{x}_{i\#}(t)$ are their associated faults. With Eq. 42, it follows from Eq. 43 that $r_i(t)$ is decoupled only from those faults in the $\Delta\mathbf{x}_{i\#}(t)$ subset which are off-loop (whose associated variable does not appear in the control equations). For the in-loop faults, the decoupling fails, even though their associated variables are eliminated from the subsystem, and from the partial PCA model describing it.

### *Illustrative example*

We will demonstrate the design and behavior of partial PCA models on the two-input two-output system discussed before (Figure 1). The structure of the submodels will be characterized by structure (incidence) matrices. In the matrix, each row corresponds to a single-equation partial PCA "submodel," and each column to a variable. In any intersection of the matrix, a "1" means a variable is present in the submodel while a "0" means it is not.

(1) *No control constraints in the training data.* There are two linear equations over the four variables (the two plant equations) so that each partial PCA model will cover three of the variables. There are four ways how three variables can be selected from four. The structure of the submodels is shown below. We also show the sole eigenvector of the residual space obtained for each partial PCA.

| Structure | | | | Residual Space | | |
|---|---|---|---|---|---|---|
| $u_1$ | $u_2$ | $y_1$ | $y_2$ | $\boldsymbol{q}_3'$ | | |
| 1 | 1 | 1 | 0 | 0.8165 | 0.4082 | −0.4082 |
| 1 | 1 | 0 | 1 | 0.7845 | 0.5883 | −0.1961 |
| 1 | 0 | 1 | 1 | 0.5345 | −0.8018 | 0.2673 |
| 0 | 1 | 1 | 1 | 0.4082 | 0.8165 | −0.4082 |

When testing the system with faulty data, one fault present at a time, the fault-responses follow the above ideal structure.

(2) *Training and testing under ratio control.* Both the training and the testing data are subject to the control relation $u_2(t)=cu_1(t)$, with $c=1.5$. Since now a total of three relations acts on the four variables, the submodels cover two variables each, in six possible combinations. The submodel structures are shown below, together with the sole eigenvector of the residual space the partial PCA yielded for each.

| Structure | | | | Residual Space | |
|---|---|---|---|---|---|
| $u_1$ | $u_2$ | $y_1$ | $y_2$ | $\boldsymbol{q}_2'$ | |
| 1 | 1 | 0 | 0 | −0.8321 | 0.5547 |
| 1 | 0 | 1 | 0 | −0.9615 | 0.2747 |
| 1 | 0 | 0 | 1 | −0.9932 | 0.1168 |
| 0 | 1 | 1 | 0 | −0.9192 | 0.3939 |
| 0 | 1 | 0 | 1 | −0.9848 | 0.1738 |
| 0 | 0 | 1 | 1 | −0.9247 | 0.3807 |

Testing with one fault at a time returns the following fault-response structure

| $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | **1** | 1 | 0 |
| 1 | **1** | 0 | 1 |
| **1** | 1 | 1 | 0 |
| **1** | 1 | 0 | 1 |
| **1** | **1** | 1 | 1 |

This does not agree with the ideal structure above; the in-line faults $\Delta u_1$ and $\Delta u_2$ cause nonzero response even if their respective variable does not appear in the partial PCA model. This is consistent with the behavior observed with the full PCA model. Note that, if the training data is generated under ratio control but the coefficient is varied (takes at least two different values), then the partial PCA models are exactly like the ones obtained with no control present.

(3) *Training and testing under feedback control.* Now both the training and the testing data are subject to the control relation $u_2(t)=K[s-y_2(t)]$, with $s=0$, $K=1$. With three relations acting on the four variables, the submodels cover two variables each, in six possible combinations. The submodel structures are shown below, together with the eigenvector of the residual space

| Structure | | | | Residual Space | |
|---|---|---|---|---|---|
| $u_1$ | $u_2$ | $y_1$ | $y_2$ | $\mathbf{q}_2'$ | |
| 1 | 1 | 0 | 0 | −0.7071 | −0.7071 |
| 1 | 0 | 1 | 0 | −0.7071 | 0.7071 |
| 1 | 0 | 0 | 1 | −0.7071 | 0.7071 |
| 0 | 1 | 1 | 0 | −0.7071 | −0.7071 |
| 0 | 1 | 0 | 1 | −0.7071 | −0.7071 |
| 0 | 0 | 1 | 1 | −0.7071 | 0.7071 |

Testing with one fault at a time returns the following fault-response structure

| $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|
| 1 | 1 | 0 | **1** |
| 1 | **1** | 1 | **1** |
| 1 | **1** | 0 | 1 |
| 0 | 1 | 1 | **1** |
| 0 | 1 | 0 | 1 |
| 0 | **1** | 1 | 1 |

The in-line faults $\Delta u_2$ and $\Delta y_2$ cause nonzero response even if their respective variable does not appear in the partial PCA model. Again, this is consistent with the behavior observed with the full PCA model. However, if the training data is generated under feedback control, but the set point is varied (takes at least two different values) then the partial PCA models are exactly like the ones obtained with no control present.

### *Relationship to diagnosis based on explicit models*

The foregoing results are consistent with our earlier understanding of the behavior of explicit models in the context of identification and diagnosis. Explicit models are those showing the plant outputs as explicit functions of the plant inputs. Such models may be obtained from first-principle knowledge, or from empirical data by some means of systems identification. The latter usually rely on least-squares (LS) based parameter estimation. In contrast, principal component (PC) models may be considered implicit. The PC model in the residual space, consisting of the minor components, includes the information concerning the plant. The PC model in the representation space, with the major components, contains additional information concerning the normal range of the variables.

In the following paragraphs, we ignore the effects of noise, disturbances, and mismatch of the model structure. When the empirical data is fully exciting, that is, no linear relations exist among the variables, other than the relations internal to the plant, then the LS identification of the explicit model is possible. Also, in this case the PC model is "tight"; the dimension of the residual space in the PC model equals the number of plant outputs. Now, the minor components of the PC model uniquely describe the explicit model; the latter may be computed from the former (from Eq. 25, with Eq. 6).

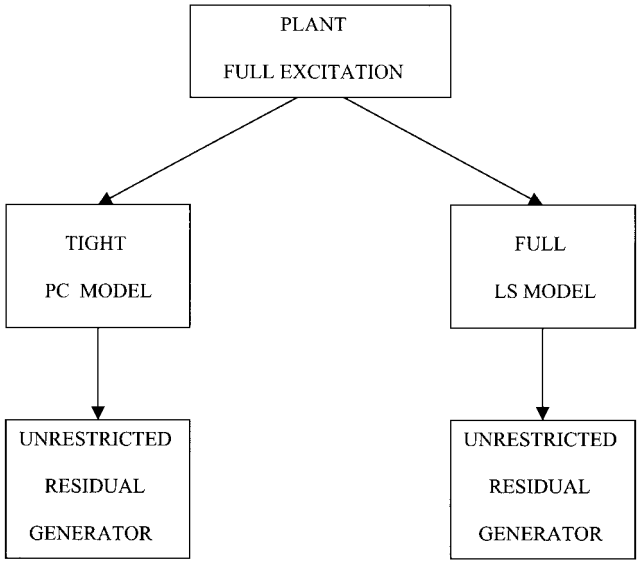


**Figure 5. Modeling and residual generation under full excitation.**

In general, all sensor and actuator faults are isolable.

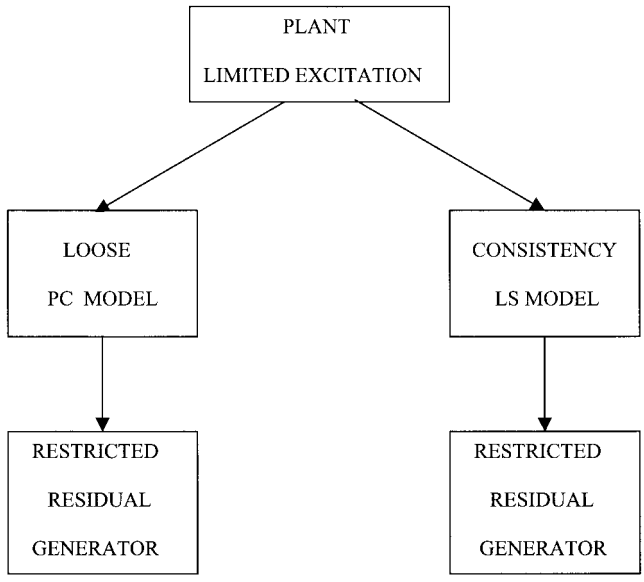


**Figure 6. Modeling and residual generation under limited excitation (linearly dependent plant variables).**

In-loop sensor and actuator faults are not isolable.

When there are linear relations among the variables that are external to the plant, then the explicit model is not identifiable. It is possible, though, to create a "consistency model" among the plant inputs and outputs; in this model, some parameters are chosen arbitrarily while the rest are obtained by LS estimation. Such a model is not a correct description of the plant, but it leads to a valid relationship among the variables, as long as the linear relations present in the original data do not change (Gertler, 1998, Chapter 12). The PC model still exists in this case, but now it is "loose"; the dimension of the residual space is higher than the number of plant outputs. Therefore, the PC model in this case does not uniquely define the explicit model either.

Analytical redundancy type fault diagnosis may be performed equally on the basis of the LS model, or of the minor component part of the PC model. If there are no external linear relations among the variables, then the LS model is the real plant model and the PC model is tight. With either of them, fault diagnosis is unrestricted, all sensor and actuator faults may, in general, be isolated (Figure 5). If there are linear relations in the data then the LS model is just a consistency model and the PC model is loose. With either of them, fault diagnosis is restricted, in the sense that off-loop sensor and actuator faults may be isolated, but in-loop faults may not (Figure 6). For the PC model, we have shown this in this article. For the LS model, the same result was obtained earlier (Gertler, 1998, Chapter 12).

In summary, analytical redundancy type fault diagnosis places the same requirements on the process data used for model building, whether the model is obtained by LS identification or by PC transformation. In either case, full excitation is needed for unrestricted fault isolation. In either case, the presence of linear relations in the data restricts the isolation of faults, but only of faults of those devices that are involved in control loops. And in either case, it only takes some variations in the control relations to remove those restrictions.

## Discrete Dynamic Systems

Dynamic systems are naturally described by differential equations. For computer-related processing, differential equations are replaced (approximated) by difference equations which act on discrete samples of the variables. The difference equation models usually assume constant signals between the sampling instants.

In the difference equation model, the variables appear with a number of past samples, in addition to the present one. If past samples are there only for the output variables, the model is autoregressive (AR). If only the input variables have their past samples, the model is moving average (MA). If both, the model is autoregressive–moving average (ARMA).

In PCA modeling, past samples pose as "pseudo variables." Pseudo variables in plant models increase the number of variables without changing the number of equations. Past samples may also appear in control relations; these have to be taken into account when determining whether a control relation poses a linear equation on the variables.

The behavior of AR and MA systems will be explored below. ARMA systems may then be explained as a combination of the two.

### *Autoregressive discrete plants*

An autoregressive difference equation for a single plant output is

$$y_i^0(t) = \sum_{j=1}^{k} a_{ij} u_j^0(t) - \sum_{g=1}^{\nu} d_{ig} y_i^0(t-g) \tag{44}$$

where $\nu$ is the dynamic order of the system. For a set of outputs $\mathbf{y}^0(t)$, the autoregressive plant equations are

$$\mathbf{y}^0(t) = \mathbf{A}\mathbf{u}^0(t) - \sum_{g=1}^{\nu} \mathbf{D}_g \mathbf{y}^0(t-g) \tag{45}$$

where $\boldsymbol{D}_g$, $g = 1 \cdots \nu$, are diagonal matrices. Equivalently

$$\mathbf{B}\mathbf{x}^0(t) = \mathbf{0} \tag{46}$$

where now

$$\mathbf{B} = [\mathbf{A} \quad -\mathbf{I} \quad -\mathbf{D}_1 \cdots -\mathbf{D}_\nu]$$

$$\mathbf{x}(t) = [\mathbf{u}'(t) \quad \mathbf{y}'(t) \quad \mathbf{y}'(t-1) \cdots \mathbf{y}'(t-\nu)]' \tag{47}$$

*Full PCA.* In the full PCA model, there are now $k + m(\nu + 1)$ variables, $k$ inputs and $m$ outputs, the latter with $\nu + 1$ samples each. The dimension of the residual space is still $m$. In the $\boldsymbol{Q}_R'$ matrix, there is a separate column for each sample of each output. However, the columns belonging to different samples of the same output, that is, to $y_i(t)$, $y_i(t-1) \cdots y_i(t-\nu)$, are co-linear. To see this, consider the transformation (26) $\boldsymbol{Q}_R' = \boldsymbol{M}\boldsymbol{B}$, where $\boldsymbol{M}$ is a full-rank square matrix. Clearly, $-\boldsymbol{M}$ is the part of $\boldsymbol{Q}_R'$ which belongs to $\mathbf{y}(t)$ while $\boldsymbol{M}\boldsymbol{D}_g$, $g = 1 \cdots \nu$, are those belonging to $\mathbf{y}(t-g)$. Since the $\boldsymbol{D}_g$ matrices are diagonal, any column of any $\boldsymbol{M}\boldsymbol{D}_g$ is co-linear with the respective column of $\boldsymbol{M}$. This is important because

- The responses to the faults associated with $y_i(t)$, $y_i(t-1) \cdots y_i(t-\nu)$, that is, to the consecutive samples of $\Delta y_i(t)$, have the same direction in the residual space;
- A projection of the residual which decouples from $\Delta y_i(t)$ will decouple also from $\Delta y_i(t-g)$, $g = 1 \cdots \nu$.

*The effect of integration.* An integrator in the plant model,

$$y_i^0(t) = \sum_{j=1}^{k} a_{ij} u_j^0(t) + y_i^0(t-1) \tag{48}$$

is a special autoregressive system. Now the coefficient of $y_i^0(t-1)$ is 1, implying that the column in $\boldsymbol{Q}_R'$ which belongs to $y_i(t-1)$ is not only co-linear with the one belonging to $y_i(t)$, but it is its exact negative. Thus, if $\Delta y_i(t-1) = \Delta y_i(t)$, that is, if the fault is constant, its effect is canceled in the residual; a step-fault is noticeable only in one residual sample. Further, in the full autoregressive system (Eq. 44), if the autoregression contains integration, then

$$1 + \sum_{g=1}^{\nu} d_{ig} = 0 \tag{49}$$

Thus, the sum of the columns in $\boldsymbol{Q}_R'$ belonging to $y_i(t) \cdots y_i(t-g)$ is also zero, and the effect of a constant fault $\Delta y_i$ does not show up in the residual. A step-fault in this case is noticable in $\nu$ samples.

*Partial PCA.* When designing structures for partial PCA, we seek subsystems, characterized by a single equation each, which are defined by the transformation $\mathbf{v}_i' B x^0(\tau) = 0$. The vector $\mathbf{v}_i'$ is selected so that it is orthogonal to certain columns of $\boldsymbol{B}$. It follows from the structure of the $\boldsymbol{B}$ matrix that if $\mathbf{v}_i'$ is chosen to be orthogonal to a column in $\boldsymbol{I}$, it is orthogonal also to the respective column of $\boldsymbol{D}_g$, $g = 1 \cdots \nu$. Thus, a partial PCA model either contains an output $y_i(t)$ with all its samples $y_i(t) \cdots y_i(t-\nu)$, or it does not contain it at all. This, of course, is consistent with the behavior of the full PCA described above.

### *Moving average discrete plants*

A moving average difference equation for a single plant output is

$$y_i^0(t) = \sum_{j=1}^{k} \sum_{g=1}^{\nu} a_{ijg} u_j^0(t-g) \tag{50}$$

where $\nu$ is the dynamic order of the system. For a set of outputs $\mathbf{y}^0(t)$, the moving average plant equations are

$$\mathbf{y}^0(t) = \sum_{g=0}^{\nu} \mathbf{A}_g \mathbf{u}^0(t-g) \tag{51}$$

where $\boldsymbol{A}_g$, $g = 0 \cdots \nu$, are in general *not* diagonal matrices. Equivalently

$$\mathbf{B}\mathbf{x}^0(t) = \mathbf{0} \tag{52}$$

where now

$$\mathbf{B} = [\mathbf{A}_0 \cdots \mathbf{A}_\nu \quad -\mathbf{I}]$$

$$\mathbf{x}(t) = [\mathbf{u}'(t) \quad \mathbf{u}'(t-1) \cdots \mathbf{u}'(t-\nu) \quad \mathbf{y}'(t)]' \tag{53}$$

*Full PCA.* There are now $m + k(\nu + 1)$ variables, including the $\nu$ shifted samples of each of the $k$ inputs, and $m$ equations. The responses to the faults associated with the various samples of $u_j(t)$, that is, to $\Delta u_j(t) \cdots \Delta u_j(t-\nu)$, are not co-linear. The combined response points in the direction of a weighted resultant; its direction is constant if the fault is constant but varies with time if the fault does.

One may decouple the residual from a constant fault by projecting it onto a subspace orthogonal to the resultant direction. If, however, the fault may vary with time, decoupling requires the augmentation of the equation set with time-shifted equations. While one may decouple from $m-1$ faults using $m$ equations in a static framework, in a moving average system one needs $\nu(m-1)$ time-shifted versions of each of the $m$ equations, in addition to the original one, to decouple from the present and $\nu$ past samples of each of the $m-1$ faults. The logic of this will be explained below, in connection with the partial PCA design. Technically, the augmented PCA is performed by including $\boldsymbol{y}(t) \cdots \boldsymbol{y}[t-\nu(m-1)]$ and $\boldsymbol{u}(t) \cdots \boldsymbol{u}(t-\nu m)$ in the dataset. Such an augmented PCA is, of course, quite complex.

*Partial PCA.* We are seeking a subsystem, obtained as $\mathbf{v}_i'\boldsymbol{B}\boldsymbol{x}^0(\tau) = 0$, which is decoupled from all samples of $m-1$ selected inputs. In general, this cannot be satisfied with a $\boldsymbol{v}_i'$ vector $m$ long. Assuming that all the decoupled variables are present in all equations with samples $g = 0 \cdots \nu$, this represents $(m-1)(\nu+1)$ orthogonality conditions. Adding a set of time-shifted equations $\boldsymbol{B}\boldsymbol{x}^0(t-1) = \mathbf{0}$ would increase the length of $\mathbf{v}_i'$ to $2m$ and the number of orthogonality conditions to $(m-1)(\nu+2)$. With $g = 0 \cdots \nu$ across the board, the possibilities catch up with the requirements when there are $\nu(m-1)$ time-shifted equation sets, that is, up to $\boldsymbol{B}\boldsymbol{x}^0[t-\nu(m-1)] = \mathbf{0}$; now the length of $\boldsymbol{v}_i'$ is $(m-1)\nu m + m$, while the number of conditions is $(m-1)\nu m + m - 1$. Note, however, that due to interrelations among the input-output equations, the order of the 1-D subsystems never exceeds the (McMillan) degree of the underlying multivariable system (Gertler, 1998).

If not all decoupled variables are present in all equations with full order, then the number of necessary time-shifts is reduced. To obtain the exact partial PCA models describing these subsystems, one needs to formally perform the algebraic elimination. This procedure may be simplified significantly by the use of shift operator (transfer function) notation, not discussed in this article (Gertler and Singer, 1990). Alternatively, approximate subsystem models may be found empirically, by starting with a first-order partial model and increasing the model order one-by-one until there is exactly one (close-to-) zero eigenvalue (Gertler et al., 2000).

### *Autoregressive-moving average discrete plants*

The ARMA plant model is the combination of the AR and MA models

$$\mathbf{y}^0(t) = \sum_{g=0}^{\nu} \mathbf{A}_g \mathbf{u}^0(t-g) - \sum_{g=1}^{\nu} \mathbf{D}_g \mathbf{y}^0(t-g) \tag{54}$$

where $\boldsymbol{D}_g$, $g = 1 \cdots \nu$, are diagonal. We use the same upper limit $\nu$ for the two sums for convenience; if the actual limits are different, some of the coefficient matrices in Eq. 54 are zero. In PCA modeling of ARMA systems, the AR terms behave as the terms of the AR model (Eq. 45) and the MA terms as those in the MA model (Eq. 51).

### *Illustrative example—Autoregressive system*

Consider a system

$$y_1(t) = 0.2u_1(t) + 0.1u_2(t) + 0.8y_1(t-1)$$

$$y_2(t) = 0.4u_1(t) + 0.3u_2(t) + 0.9y_2(t-1) \tag{55}$$

This system is autoregressive. The number of (pseudo-) variables is six. With two linear relations, the model-space is 4-D, while the residual-space is 2-D. Generating data in open loop, the residual space was obtained as

| | $u_1(t)$ | $u_2(t)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
|---|---|---|---|---|---|---|
| $\mathbf{q}_5'$ | 0.3105 | 0.2143 | 0.2975 | 0.5313 | −0.3719 | −0.5904 |
| $\mathbf{q}_6'$ | −0.0139 | −0.0442 | 0.5399 | −0.3350 | −0.6749 | 0.3723 |
| $q_{6j}/q_{5j}$ | −0.0448 | −0.2062 | 1.8149 | −0.6306 | 1.8149 | −0.6306 |

We show in the table also the (pseudo-) variables to which the various columns belong. As it can be seen, the predicted response direction to $y_1(t-1)$ is the same as to $y_1(t)$, and to $y_2(t-1)$ it is the same as to $y_2(t)$. With simulated constant faults, the actual response directions were obtained as

| | $\Delta u_1(t)$ | $\Delta u_2(t)$ | $\Delta y_1(t-1) = \Delta y_1(t)$ | $\Delta y_2(t-1) = \Delta y_2(t)$ |
|---|---|---|---|---|
| $e_6/e_5$ | −0.0448 | −0.2062 | 1.8149 | −0.6306 |

These agree with the predicted directions. Note that this result would be the same even if the faults were time-varying.

We will demonstrate now the design of structured partial PCAs for this system. With four physical variables and two linear relations, the possible submodel structures are as shown below. The structures $S_1$ and $S_2$ are the original single-output subsystems; here the total number of (pseudo-) variables in the submodel is four. The structures $S_3$ and $S_4$ involve the elimination of one input each, at the expense of including both outputs (and their autoregressive term); now the number of (pseudo-) variables is five. The residual space in all cases is 1-D. We show below the single eigenvector obtained in simulation for each case; a blank in the vector indicates that the concerned (pseudo-) variable is not present in the submodel.

| | structure | | | | residual space $\mathbf{q}'_R$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $u_1$ | $u_2$ | $y_1$ | $y_2$ | $u_1(t)$ | $u_2(t)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
| $S_1$ | 1 | 1 | 1 | 0 | −0.1538 | −0.0769 | −0.6154 | | 0.7692 | |
| $S_2$ | 1 | 1 | 0 | 1 | −0.2787 | −0.2090 | | −0.6271 | | 0.6967 |
| $S_3$ | 1 | 0 | 1 | 1 | 0.0491 | | 0.5889 | −0.2208 | −0.7361 | 0.2454 |
| $S_4$ | 0 | 1 | 1 | 1 | | 0.0345 | −0.5527 | 0.3109 | 0.6909 | −0.3454 |

Testing with one fault at a time, the partial PCA residuals behaved according to the ideal structure.

### *Illustrative example—autoregressive moving average system*

Now consider the system

$$y_1(t) = 0.2u_1(t) + 0.1u_2(t) + 0.3u_2(t-1) + 0.8y_1(t-1)$$

$$y_2(t) = 0.4u_1(t) + 0.3u_2(t) + 0.2u_2(t-1) + 0.9y_2(t-1) \tag{56}$$

This system is autoregressive moving average. The number of (pseudo-) variables is 7. With two linear relations, the model-space is 5-D, while the residual-space is 2-D. Generating data in open loop, the residual space was obtained as

| | $u_1(t)$ | $u_2(t-1)$ | $u_2(t)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{q}'_6$ | −0.3025 | −0.2245 | −0.2086 | −0.2931 | −0.5158 | 0.3663 | 0.5731 |
| $\mathbf{q}'_7$ | 0.0242 | −0.1193 | 0.0510 | −0.5258 | 0.3502 | 0.6573 | −0.3892 |
| $q_7/q_6$ | −0.0800 | 0.5316 | −0.2446 | 1.7942 | −0.6790 | 1.7942 | −0.6790 |

Now, for the autoregressive variables, $y_1$ and $y_2$, the $y_1(t-1)$ column is co-linear with the $y_1(t)$ column and the $y_2(t-1)$ is co-linear with the $y_2(t)$ column, but for the moving average variable $u_2$, the $u_2(t-1)$ column is not co-linear with the $u_2(t)$ column. This means that in response to a time-varying fault $\Delta u_2(t)$, the direction of the residual response would vary with time. If, however, $\Delta u_2$ is constant then the response points in the resultant direction of the $\Delta u_2(t-1)$ and $\Delta u_2(t)$ responses, which is

$$q_7/q_6 = (-0.1193 + 0.0510)/(-0.2245 - 0.2086) = 0.1577$$

Testing with constant faults has produced the expected response directions as follows

| | $\Delta u_1(t)$ | $\Delta u_2(t-1) = \Delta u_2(t)$ | $\Delta y_1(t-1) = \Delta y_1(t)$ | $\Delta y_2(t-1) = \Delta y_2(t)$ |
|---|---|---|---|---|
| $e_7/e_6$ | −0.0800 | 0.1578 | 1.7942 | −0.6790 |

When designing a structured set of partial PCA models for this system, the Boolean structure of the submodels is the same as in the case of System 55. Submodels $S_1$ and $S_2$ are again parts of the original system. Submodel $S_4$ is obtained by the elimination of the $u_1$ variable; since it is present only as $u_1(t)$, the resulting subsystem is not different from the earlier case. However, $u_2$ is present as $u_2(t)$ and $u_2(t-1)$; to eliminate this variable, one needs to include a single time-shifted equation set (with $\nu = 1$ and $m = 2$, $\nu(m-1) = 2$). Thus, $S_3$ will contain, in addition to the (pseudo-) variables present in the original model, also $u_1(t-1)$, $y_1(t-2)$ and $y_2(t-2)$. The submodels obtained by simulation are shown below.

| | Structure | | | | Residual Space $\mathbf{q}'_R$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $u_1$ | $u_2$ | $y_1$ | $y_2$ | $u_1(t)$ | $u_2(t-1)$ | $u_2(t)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
| $S_1$ | 1 | 1 | 1 | 0 | 0.1499 | 0.2249 | 0.0750 | 0.5996 | | −0.7495 | |
| $S_2$ | 1 | 1 | 0 | 1 | 0.2760 | 0.1380 | 0.2070 | | 0.6211 | | −0.6901 |
| $S_3$ | 1 | 0 | 1 | 1 | ---------- see below ---------- | | | | | | |
| $S_4$ | 0 | 1 | 1 | 1 | | −0.1369 | 0.0342 | −0.5475 | 0.3080 | 0.6844 | −0.3422 |

| $S_3$ | $u_1(t-1)$ | $u_1(t)$ | $y_1(t-2)$ | $y_2(t-2)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
|---|---|---|---|---|---|---|---|---|
| | −0.4192 | −0.0399 | 0.5988 | −0.1996 | 0.1597 | −0.3194 | 0.5389 | −0.0798 |

Experiments with faults, applied one at a time, yielded the expected response structure.

## Control Constraints in Dynamic Systems

The ratio control relation $u_j(t) = cu_i(t)$ and the proportional feedback relation $u_j(t) = K_i[s_i(t) - y_i(t)]$ have no memory and, thus, act the same way in a dynamic as in a static framework. For the integrating controller, however, the dynamics need to be taken into account.

### *Integrating controller*

Consider a proportional-plus-integrating controller in the continuous time $\tau$

$$u_j(\tau) = K_{Pi}\,[s_i(\tau) - y_i(\tau)] + K_{Ii}\,J(\tau) \tag{57}$$

where

$$J(\tau) = \int_{\vartheta=0}^{\tau} [s_i(\vartheta) - y_i(\vartheta)]d\vartheta \tag{58}$$

A discrete approximation for Eqs. 57–58 is

$$u_j(t) = K_{Pi}\,[s_i(t) - y_i(t)] + K_{Ii}J(t-1) + K_{Ii}\,T\,[s_i(t) - y_i(t)] \tag{59}$$

where $t = \tau/T$ and $T$ is the sampling interval. Writing Eq. 57 for $\tau - T$ and substituting $J(t-1)$ into Eq. 59 yields

$$u_j(t) = u_j(t-1) + (K_{Pi} + K_{Ii}T)[s_i(t) - y_i(t)] - K_{Pi}[s_i(t-1) - y_i(t-1)] \tag{60}$$

Discrete PID controllers and more complex discrete control algorithms can be derived and presented in a similar way.

A discrete control algorithm represents a linear constraint over the plant variables $\boldsymbol{x}(t)$ if and only if both of the following conditions are present:

(1) The ratio coefficient $c_i$ (in ratio control) or the set point $s_i(t)$ (in feedback control) is constant;

(2) The dynamic scope of the control algorithm is entirely inside that of the plant model, that is, the control algorithm does not contain any variable sample which is not present in the plant model.

If the ratio coefficient or the set point varies while the training data is collected, the control constraint is effectively "removed," just like in the static case. Further, if the dynamic control algorithm contains variable samples not present in the plant model, then it is not a linear constraint on the plant variables.

Dynamic linear control constraints, under zero (constant) set point, may be described in general as

$$\mathbf{x}(t) = \sum_{g=0}^{\nu} \mathbf{C}_g\mathbf{x}(t-g) \tag{61}$$

### *Illustrative example*

We are demonstrating the effect of the PI controller

$$u_2(t) = u_2(t-1) - 1.2y_2(t) + 0.5y_2(t-1) \tag{62}$$

on the pure AR system (Eq. 55) and on the ARMA system (Eq. 56).

*Autoregressive System.* Observe that the AR system model (Eq. 55) does not contain the sample $u_2(t-1)$. Thus, the control Eq. 62 does not constitute a linear constraint on the plant variables. Indeed, using data obtained under control (zero set point), the PCA performed on the plant variables detected only two linear relations (the plant equations). The residual space exhibited the exact same properties as the one obtained in open loop. Similarly, the structured partial PCA models generated from the data behaved the same as their open-loop equivalents.

*Autoregressive Moving Average System.* The ARMA model (Eq. 56) contains all the four variable samples appearing in Eq. 62. Thus, the control equation is now effectively a linear relation on the plant variables, provided the set point is zero (constant).

Using data generated under control (with zero set point), the PCA detected three linear relations. The fault-response directions predicted from the residual space are

| | $u_1(t)$ | $u_2(t-1)$ | $u_2(t)$ | $y_1(t-1)$ | $y_2(t-1)$ | $y_1(t)$ | $y_2(t)$ |
|---|---|---|---|---|---|---|---|
| $q_{6j}/q_{5j}$ | −0.2847 | −0.1014 | −0.0909 | 1.3648 | −1.2582 | 1.3648 | −3.8835 |
| $q_{7j}/q_{5j}$ | 0.1531 | −3.0490 | −0.6527 | 0.9658 | 0.4687 | 0.9658 | 3.4435 |

With testing data, generated under the same feedback control, applying one fault at a time, the residuals are

| response to | $\Delta u_1$ | $\Delta u_2$ | $\Delta y_1$ | $\Delta y_2$ |
|---|---|---|---|---|
| $e_6/e_5$ | −0.2847 | 0.1681 | 1.3648 | −0.7691 |
| $e_7/e_5$ | 0.1531 | 0.3762 | 0.9658 | −0.0856 |

The variables $y_1$ and $u_1$ are not affected by feedback; the $y_1(t)$ and $y_1(t-1)$ columns still have the same slopes in $\mathbf{Q}'_R$, and the predicted directions agree with the actual fault responses. For $y_2$ and $u_2$, however, this is not the case; the $y_2(t)$ and $y_2(t-1)$ columns have different slopes, and the actual fault responses for $y_2$ and $u_2$ are different from those predicted from $\mathbf{Q}'_R$.

With three equations over four variables, six submodels can be generated, each containing just two variables. Each submodel consists of a single equation. The elimination results in various dynamic orders for the remaining variables, ranging from 1st to 3rd, with or without integration. The residuals were tested with single faults. Because of the integrating effect of the control equation, some of the residuals did not respond to some faults with constant value. Therefore, the test was performed with time-varying faults; the results are shown below.

| Subsystem | Ideal structure | | | | Actual structure | | | |
|---|---|---|---|---|---|---|---|---|
| | $u_1$ | $u_2$ | $y_1$ | $y_2$ | $u_1$ | $u_2$ | $y_1$ | $y_2$ |
| $S_1$ | 1 | 1 | 0 | 0 | 1 | 1 | 0 | **1** |
| $S_2$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $S_3$ | 1 | 0 | 0 | 1 | 1 | **1** | 0 | 1 |
| $S_4$ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | **1** |
| $S_5$ | 0 | 1 | 0 | 1 | 0 | **0** | 0 | **0** |
| $S_6$ | 0 | 0 | 1 | 1 | 0 | **1** | 1 | 1 |

Observe that only $S_2$ returns the correct structure; in this subsystem, only the variables not affected by control are present. The subsystem $S_5$ is, in fact, the controller equation which is completely insensitive to any fault. The remaining four subsystems each are supposed to be decoupled from one of the two variables affected by control; because of the cross-effect imposed by the control relation, the decoupling is not perfect.

## Conclusion

In direct model-based diagnostic approaches, such as consistency relations using explicit models or the minor components of a PCA model, the nominal system model is applied to the observed variables. The model equations of the plant proper link the true values of the variables; if the observations differ from the true values, then nonzero residuals arise which can then be utilized for the diagnosis of sensor and actuator faults. However, model equations describing controller action link the observations rather than the true variables; if these equations are used as if they were plant equations, then mis-isolation may result. While, in first-principle modeling, control equations are clearly separated from plant equations, and, in least-squares-type model building, control equations may cause nonidentifiability, PCA modeling does not distinguish those equations from the equations of the plant.

In this article, we provide a detailed analysis of how control equations affect analytical redundancy based diagnosis that uses PCA models. We investigate ratio and feedback control in linear static and discrete dynamic systems, and we consider both full and partial PCA models. We show that all that it takes to eliminate the adverse effects is to vary the control set point (in feedback control) or the ratio coefficient (in ratio control) in the course of collecting the training data.

Analytical redundancy methods provide powerful tools for fault isolation, but they require sufficient excitation in the model-building phase. This has been shown equally true whether the model of the plant is based on the minor components of a PCA model or is obtained by least-squares type systems identification. Other PCA-based methods may not require sufficient excitation, but may offer less convenient isolation tools. It is a challenging subject for further research to compare these methods, theoretically and on benchmark problems, to evaluate their respective advantages and disadvantages and the tradeoffs among them.

## Acknowledgments

This work has been supported by NSF under Grant No. ECS-9906250. A portion of this material has been presented at the IFAC Safeprocess 2003 Symposium (Gertler and Cao, 2003).

## Literature Cited

Chow, E. Y., and A. S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems," *IEEE Tr. Auto. Control,* **AC-29,** 603 (1984).

Dunia, R., S. J. Qin, T. F. Edgar, and T. J. McAvoy, "Identification of Faulty Sensors using Principal Component Analysis," *AIChE J.,* **42,** 2797 (1996).

Dunia, R., and S. J. Qin, "Joint Diagnosis of Process and Sensor Faults using PCA," *Control Eng. Practice,* **6,** 457 (1998).

Frank, P., "Fault Diagnosis in Dynamic Systems using Analytical and Knowledge-based Redundancy," *Automatica,* **26,** 459 (1990).

Gertler, J., and D. Singer, "A New Structural Framework for Parity

Equation Based Fault Detection and Isolation," *Automatica,* **26,** 381 (1990).
Gertler, J., and T. J. McAvoy, "Principal Component Analysis and Parity Relations—A Strong Duality," *IFAC Safeprocess Symp.,* **2,** 837 (1997).
Gertler, J., *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, New York (1998).
Gertler, J., W. Li, Y. Huang, and T. J. McAvoy, "Isolation Enhanced Principal Component Analysis," *AIChE J.,* **45,** 323 (1999).
Gertler, J., Y. Hu, Y. Huang, and T. J. McAvoy, "Structured Partial PCA: Extension to Polynomial Nonlinearities," *Preprints of IFAC Safeprocess Symp.,* **2,** 1004 (2000).
Gertler, J., "All Linear Methods are Equal—and Extendible to (some) Nonlinearities," *Int. J. of Robust and Nonlinear Control,* **12,** 629 (2002).
Gertler, J., and J. Cao, "PCA-Based Process Diagnosis in the Presence of Control," *Preprints of IFAC Safeprocess Symp.,* 849 (2003).
Jones, H. L., "Failure Detection in Linear Systems," PhD Diss., Dept. Aero. and Astro, Massachusetts Inst. of Technology (1973).
Kourti, T., and J. F. MacGregor, "Process Analysis, Monitoring and Diagnosis using Multivariate Projection Methods," *Chemometrics and Intelligent Laboratory Systems,* **28,** 3 (1995).
Kourti, T., and J. F. MacGregor, "Multivariate SPC Methods for Process and Product Monitoring," *J. of Quality Technology,* **28,** 409 (1996).
MacGregor, J. F., and T. Kourti, "Statistical Process Control of Multivariate Processes," *Control Eng. Practice,* **3,** 403 (1995).
Negiz, A., and A. Cinar, "Statistical Monitoring of Multivariable Dynamic Processes with State-Space Models," *AIChE J.,* **43,** 2002 (1997).
Piovoso, M., K. Kosanovich, and P. Pearson, "Monitoring Process Performance in Real Time," *American Control Conf.,* 2359 (1992).
Raich, A., and A. Cinar, "Diagnosis of Process Disturbances by Statistical Distance and Angle Measures," *Comput. and Chem. Eng.,* **21,** 661 (1997).
Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica,* **12,** 601 (1976).
Wise, B., and N. L. Ricker, "Feedback Strategies in Multiple Sensor Systems," *AIChE Symp. Series,* **85,** 19 (1991).
Yoon, S., and J. F. MacGregor, "Statistical and Causal Model-Based Approaches to Fault Detection and Isolation," *AIChE J.,* **46,** 1813 (2000a).
Yoon, S., and J. F. MacGregor, "Fault Diagnosis with Multivariate Statistical Models. Part I: Steady-State Fault Signatures," *J. of Process Control,* **11,** 387 (2000b).

*Manuscript received Nov. 11, 2002, and revision received Jun. 13, 2003.*