ROBUST ISOLATION OF SENSOR FAILURES

R. Xu and C. Kwan

ABSTRACT

Sensor self-validity check is a critical step in system control and fault diagnostics. In this paper, a robust approach to isolate sensor failures is proposed. First, a residual model for a given system is built off-line and directly based on input-output measurement data. The residual model outputs are called “primary residuals” and are zero when there is no fault. Most conventional approaches to residual model generation are indirect, as they first require the determination of state-space or other models using standard system identification algorithms. Second, a new max-min design of structured residuals, which can maximize the sensitivity of structured residuals with respect to sensor failures, is proposed. Based on the structured residuals, one can then isolate the sensor failures. This design can also be done in an off-line manner. It is an optimization procedure that avoids local optimal solutions. Simulation and experimental results demonstrated the effectiveness of the proposed method.

KeyWords: Residual model, robust fault isolation, sensor validation, max-min design, fault detection and isolation.

I. INTRODUCTION

The ability to detect faulty sensors is important in many diverse applications. There are usually four types of sensor failures: bias, precision degradation, complete failure, and drift. These sensor faults may have different impacts on overall system performance depending upon the applications. In feedback control applications, faulty sensors give wrong information about the system status, which could cause disastrous results as system may go unstable. Even the system is stable, inaccurate sensor values may introduce poor regulation or tracking performance, which may be highly undesirable for many high precision control applications. In fault detection and isolation (FDI) of system faults, faulty sensor outputs will also cause inaccurate diagnostic results or false alarms.

As was described in a recent paper [13], the work on sensor validation can be classified into two categories: gross error detection in sensor reconciliation and sensor fault detection. Representative work in sensor reconciliation includes [3,12,14]. In the area of sensor fault detection, model based FDI techniques surveyed by Gertler [6] and Frank [4] can be applied. Several recent papers investigate new FDI methods in automotive engines, hard disk drives, and semiconductor process monitoring [9,10,15]. Almost all conventional methods in sensor FDI [1,4,5,6,8,9,10,15], except some methods using Principal Component Analysis (PCA) [7,13,16], require the determination of explicit state-space or transfer function models. A typical discrete state-space model is of the form

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k + Du_k$$

where $u_k$, $x_k$, $y_k$ denoting input, state, and output sequences, respectively. The model parameters are termed as $(A, B, C, D)$. Based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated. The approach in this paper completely eliminates the determination of state-space models $(A, B, C, D)$ or transfer functions, which means our method is based on the identified model parameters, a residual model is then generated.

In this paper, a robust approach to sensor fault isolation is proposed. The method consists of two steps. First, a residual model of the process is constructed by directly using input-output measurements. The idea was motivated by recent advances in system identification developments, especially the techniques of Errors-in-Variables (EIV) subspace approach [2,11]. Most conventional approaches to residual model generation are indirect in nature, as they first require the determination of
state-space or other models using standard system identification algorithms. And then the residual model is generated based on the identified models. Comparing with conventional methods, the current approach is direct, avoids the inherent numerical and estimation errors in building the state-space model of the process, and hence is more accurate. The outputs of the residual model are called “primary residuals” and can be used to detect the presence of sensor failures. However, primary residuals cannot pinpoint the particular faulty sensor. Additional processing to generate signals known as “structured residuals” is needed. As the isolation of sensor failures directly relates to the sensitivity of structured residuals, the structured residual generation is a very critical step in sensor failure isolation. Second, a systematic max-min design procedure to generate structured residuals is presented here. The main idea is to maximize the sensitivity of structured residuals with respect to sensor failures. The design has a Monte-Carlo like search procedure to determine a near optimal solution. It is worth mentioning that local optimal solutions can be avoided. Comparing to a recent work known as Structured Residual Approach with Maximized Sensitivity (SRAMS) [13], the method in this paper has better performance in both simulations and experimental verifications. It is worth to emphasize that multiple faults can be handled very easily in the proposed design framework.

The paper is organized as follows. Section 2 will review background and notations. The technical details will be introduced here. Section 3 describes an experimental setup, which will be used to demonstrate the performance of the proposed method. Section 4 summarizes both simulations and experimental studies. A comparative study is carried out here. Finally, conclusions and suggestions for future research are included in Section 5.

II. SENSOR FAILURE ISOLATION THEORY

2.1 Background and notation

Although the proposed method can be used for both static and dynamic systems, we use a static system to illustrate background and notation of the proposed method. Suppose the sensor measurements are denoted as

\[ z(t) = z^*(t) + \Delta z(t) \]  

(1)

with \( z^*(t) \) the normal sensor outputs and \( \Delta z(t) \) the additive sensor faults such as bias, drifts, etc. Then we call the following parity relation a process model [6,7,13]

\[ B_r z^*(t) = 0 \]  

(3)

where \( B_r \) is called the residual model matrix and \( e(t) \) in (2) is called a primary residual vector that is computed from the measurements.

It should be noted this process model is not a model in the conventional sense. As can be seen from (2) and (3), the model output is zero when there is no fault and nonzero when there is fault. A more appropriate term for (2) is parity relation [6,7,13]. However, we will ignore this minor issue and treat “process model” and “parity relation” as the same. Details will be described in the next section for how to derive (2) by using PCA for static systems and EIV subspace method for dynamic systems. Primary residuals can be used for detecting the presence of faulty sensors. However, they do not reveal which sensor has failed. To isolate faults, we need to generate structured residuals of the form

\[ r(t) = W e(t) \]  

(4)

where \( W \) is a constant transformation matrix that is designed so that each residual is sensitive to a particular subset of faults but not sensitive to the other faults. To determine a suitable \( W \), an incidence matrix is first constructed. In the incidence matrix, the rows are the residual structures while the columns are the fault modes. An \( I \) element means that the concerned residual does respond to the concerned fault while an \( O \) element means it does not. An incidence matrix structure is termed “isolating” if each column is different and “strongly isolating” if the columns remain different from other columns when \( I \)s are turned into \( O \)s. One example of strongly isolating structure is shown in Table 1. If \( z_1 \) has a fault, \( \Delta z_1 \) is nonzero. Then the residuals \( r_1, r_2, \) and \( r_6 \) will be zero and \( r_3, r_4, \) and \( r_5 \) will be nonzero. So one can monitor the residuals to indicate failure in \( z_1 \). Similarly, if \( z_4 \) has a fault, the residuals \( r_1, r_6, \) and \( r_7 \) will be zero and \( r_2, r_3, \) and \( r_5 \) will be nonzero. If both \( z_1 \) and \( z_4 \) fail simultaneously, the residuals \( r_1, r_2, r_3, r_4, r_5, r_6 \) will be all nonzero except \( r_7 \). Therefore, the incidence matrix can be used to clearly indicate what failures occur in a sys-

<table>
<thead>
<tr>
<th>( r_1 )</th>
<th>( r_2 )</th>
<th>( r_3 )</th>
<th>( r_4 )</th>
<th>( r_5 )</th>
<th>( r_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O )</td>
<td>( O )</td>
<td>( I )</td>
<td>( I )</td>
<td>( O )</td>
<td>( I )</td>
</tr>
</tbody>
</table>
tem. It should be noted the number of zeros in each row of Table 1 has an upper bound. If there are \( m \) independent measurements, one can assign at most \( m-1 \) zeros in each row. Consequently, one can detect up to \( m-1 \) simultaneous multiple faults.

When structured residuals are generated by (4) then the \( i^{th} \) row of \( W \), \( w_i^* \), is required to satisfy

\[
w_i^T B_i^t = 0
\]

where \( B_i^t \) contains those columns of \( B_i \) matrix in (2) that belong to the faults assigned for zero response in the \( i^{th} \) residual structure. Conditions for guaranteeing the existence of solution for \( W \) were detailed in [7].

### 2.2 Residual model generation

Here we describe methods to derive the residual model mentioned in (2). Although we call “residual model” a model, it is not a dynamic model in the conventional sense. The residual model is a parity relation describing the relationship of various quantities in the system whereas dynamic models such as state-space or transfer function models describe the dynamics of the system. In other words, dynamic models can generate time domain input-output responses of a system and the residual models give zero outputs when system has no faults and nonzero outputs when there is fault.

#### Static systems

The normal process model (2) can be obtained by using PCA [7,13]. We summarize the theory here. The normal sensor values \( \ast(*)\) of sensor measurements, is assumed to be stationary and can be decomposed into

\[
z^* (t) = P v + \tilde{P} \nu
\]

where \( P \in R^{n(n-n)} \) are orthogonal eigenvectors associated with the principal eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-n} \) of the correlation matrix, defined as \( E(z^* z^*) \) (\( z^* = E(z) \), of \( z(t) \)). \( \tilde{P} \) denotes the remaining eigenvectors associated with the minor eigenvalues \( \lambda_{n-n+1}, \ldots, \lambda_{n} \) (zero when there is no noise in the system). As the system is assumed to be stationary, the correlation matrix is constant. The vectors \( v \) and \( \nu \) are the principal components and residual components, respectively. Since \( P \) and \( \tilde{P} \) are orthogonal, premultiplying (6) by \( \tilde{P}^T \) leads to

\[
\tilde{P}^T z^* (t) = \tilde{P}^T P v = \tilde{v}
\]

Comparing (7) with (2), the residual model is a constant matrix given by

\[
B_v = \tilde{P}^T.
\]

#### Dynamic systems

In the previous section, we described residual model generation for static linear systems using PCA. Here we describe a general theory to generate the residual model for linear dynamic systems. The key idea was motivated by EIV-subspace identification method [2,11].

Assume the system to be considered is given as follows.

\[
x_{k+1} = Ax_k + B \tilde{u}_k + p_k
\]

\[
y = C x_k + D \tilde{u}_k
\]

where \( \tilde{y}_k \in R^m \), \( \tilde{u}_k \in R^r \), \( x_k \in R^n \), and \( p_k \in R^o \) are noise-free outputs and inputs, states and process disturbance, respectively. \( \{A, B, C, D\} \) are system matrices with appropriate dimensions. Conventional FDI approaches requires the identification of \( \{A, B, C, D\} \) by using input-output measurements. Recent advances in system identification have proven the usefulness of a technique known as EIV-subspace identification that gives consistent estimates of \( \{A, B, C, D\} \). Although this method is more accurate than other system identification methods in determining \( \{A, B, C, D\} \), the numerical errors are still unavoidable. If we use the identified \( \{A, B, C, D\} \) to build a residual model, the numerical errors in \( \{A, B, C, D\} \) will propagate into the fault residuals and hence complicate the fault diagnosis. As will be seen shortly, a more direct and accurate procedure would be to do away with the \( \{A, B, C, D\} \) parameters and directly use some intermediate matrices in the EIV-subspace identification procedure to establish the residual model.

Our objective is to identify a residual model in the form of (2), given observations:

\[
y_k = \tilde{y}_k + q_k, \quad u_k = \tilde{u}_k + v_k
\]

where \( \{q_k\} \) and \( \{v_k\} \) are zero-mean white noise sequences. After straightforward manipulations, one gets

\[
\begin{bmatrix}
\dot{y}_{k-n} \\
y_k \\
\end{bmatrix} = \Gamma_n x_{k-n} + H_n \begin{bmatrix}
q_{k-n} \\
q_k \\
\end{bmatrix} + G_n \begin{bmatrix}
p_{k-n} \\
p_k \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
u_{k-n} \\
v_k \\
\end{bmatrix} = -H_n \begin{bmatrix}
u_{k-n} \\
v_k \\
\end{bmatrix} + \begin{bmatrix}
q_{k-n} \\
q_k \\
\end{bmatrix}
\]

Note that

\[
B_v = \tilde{P}^T.
\]
In compact form, (11) can be written as
\[ Y = \Gamma_y x_k + H_y U + G_y P - H_y V + Q \] (13)
with
\[
\begin{bmatrix}
  y_{k+1} \\
  \vdots \\
  y_k
\end{bmatrix} = \begin{bmatrix}
  u_{k+1} \\
  \vdots \\
  u_k
\end{bmatrix} - P \begin{bmatrix}
  P_{k+1} \\
  \vdots \\
  P_k
\end{bmatrix} - V \begin{bmatrix}
  V_{k+1} \\
  \vdots \\
  V_k
\end{bmatrix},
\]
and
\[
Q = \begin{bmatrix}
  q_{k+1} \\
  \vdots \\
  q_k
\end{bmatrix}.
\]
Multiplying \( \Gamma_y^{-1} \) on both sides of (13) gives
\[ \Gamma_y^{-1}(Y - H_y U) = \Gamma_y^{-1}(G_y P - H_y V + Q) \]
where \( \Gamma_y^{-1} \) is the orthogonal complement of \( \Gamma_y \). Hence, from the above equation, it can be seen that \( \Gamma_y^{-1}(Y - H_y U) \) is a zero-mean process noise. If we define a new measurement vector as
\[ Z = (Y + \Delta Y) - H_y U \] (14)
with \( \Delta Y \) denoting the sensor faults. Multiplying \( \Gamma_y^{-1} \) on (14) gives
\[ \Gamma_y^{-1}Z = \Gamma_y^{-1}(\Delta Y + Y) - \Gamma_y^{-1}H_y U = \Gamma_y^{-1}\Delta Y + \Gamma_y^{-1}(Y - H_y U) \] (15)
When there are no sensor faults, \( \Delta Y \) is zero and (15) consists of zero-mean random noise. Comparing with (2), equation (15) can be considered as the primary residual model if we define
\[ B_r = \Gamma_y^{-1}. \] (16)

Conventional approach to generate residual model is to first determine the state-space model \( \{A, B, C, D\} \) by some system identification methods. Then \( \Gamma_y \) in (12) is calculated by using the identified \( \{A, B, C, D\} \). Finally, the residual model (16) is generated. The disadvantage is that the estimation of \( \{A, B, C, D\} \) has inherent numerical and estimation errors, which will propagate to primary residuals. It would be advantageous to estimate \( \Gamma_y \) directly from the measurements. Our objective here is to determine \( B_r \) in (16) by using input-output measurement data directly. The following paragraphs describe this methodology.

Defining the following Hankel data matrix in \( \mathbb{R}^{|A||x'|+1} \)
\[
Y_{k,n+1} = \begin{bmatrix}
  y_k & y_{k+1} & \cdots & y_{k+n} \\
  \vdots & \vdots & \ddots & \vdots \\
  y_{k+n-1} & y_{k+n} & \cdots & y_{k+2n-1} \\
\end{bmatrix} \] (17)
and from (13), it turns out that
\[ Y_{k+2,n} = \Gamma_y X_{n+2,N} + H_y U_{n+2,N} + G_y P_{n+2,n} \]
\[ -H_y V_{n+2,N} + Q_{n+2,n} \] (18)
where we choose \( k = n+2 \) in (17) and with \( X_{n+2} = [x_{n+2}, \ldots, x_{n+2}] \in \mathbb{R}^{n+2} \). Denoting
\[ E_{n+2,n} = G_y P_{n+2,n} + H_y V_{n+2,n} + Q_{n+2,n} \]
and post-multiplying (18) by the instrumental variables \( [U_{1,n}^T, Y_{1,n}^T] \), we reach
\[ Y_{n+2,n} [U_{1,n}^T, Y_{1,n}^T] = \Gamma_y X_{n+2,N} [U_{1,n}^T, Y_{1,n}^T] \]
\[ + H_y U_{n+2,N} [U_{1,n}^T, Y_{1,n}^T] \]
\[ + E_{n+2,n} [U_{1,n}^T, Y_{1,n}^T] \] (19)
where as \( N \to \infty \), the last term \( E_{n+2,n} [U_{1,n}^T, Y_{1,n}^T] \) will vanish [2] because the instrumental variables \( U_{1,n}^T, Y_{1,n}^T \) do not correlate with the future events \( E_{n+2,N} \). Therefore, for large \( N \),
\[ Y_{n+2,n} [U_{1,n}^T, Y_{1,n}^T] = \Gamma_y X_{n+2,N} [U_{1,n}^T, Y_{1,n}^T] \]
\[ + H_y U_{n+2,N} [U_{1,n}^T, Y_{1,n}^T] \] (20)
Performing QR decomposition gives the following use-
We extend the method to dynamic systems, in detail for the static case. Then we briefly mention how structured residuals are acting on the primary residuals to generate the structure of Table 1, we choose a matrix of the form in Table 1 is satisfied. The matrix H is acting on the primary residuals to generate the structure of (21) into (20) produces

\[
R_{21}Q_1 + R_{22}Q_2 = \Gamma_eX_{\infty,2}X_{\infty,1}^T + H_eR_{11}Q_1.
\]

Since \(Q_1\) and \(Q_2\) are orthogonal by nature of the QR decomposition, we postmultiply \(Q_1^T\) to (22) to yield

\[
R_{22} = \Gamma_eX_{\infty,2}X_{\infty,1}^T + H_eR_{11}.
\]

And we postmultiply \(Q_2^T\) to (22) to give

\[
R_{21} = \Gamma_eX_{\infty,2}X_{\infty,1}^T + H_eR_{11}Q_1.
\]

Applying Singular Value Decomposition (SVD) on \(R_{22}\) yields

\[
R_{22} = [\hat{\Gamma}_n]_a \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix} \Gamma_n^T
\]

where \(\hat{\Gamma}_n \in R^{m \times p}\) is the consistent estimate of \(\Gamma_n\), and \(\Gamma_n^T \in R^{m \times (m-1)\times p\times (m+1)}\) is the consistent estimate of \(\Gamma_n^T\). [2].

Since \(B_eZ = B_e(Y + \Delta Y - H_e U)\), we also need to determine \(\hat{\Gamma}_n^+H_n^+\) in order to generate the primary residual vector \(B_eZ\). Multiplying \(\hat{\Gamma}_n^T\) to (24) yields

\[
\hat{\Gamma}_n^+H_n^+ = \Gamma_n^+R_{22}R_{11}^{-1}.
\]

Since we have \(\hat{\Gamma}_n^+R_{22} = \Gamma_n^+R_{22}^+\). Hence we have \(\hat{\Gamma}_n^+H_n^+ = \Gamma_n^+R_{22}R_{11}^{-1}\), where \(\Gamma_n^+\) denotes the pseudo-in-verse of \(\Gamma_n\). Note that both \(R_{11}^{-1}\) and \(R_{11}\) can be obtained from (21).

Now we have consistent estimates of both \(\hat{\Gamma}_n^+\) and \(\Gamma_n^+H_n^+\) as they are given by

\[
\text{Consistent estimate of } \Gamma_n^+ = \hat{\Gamma}_n^+ = \Gamma_n^+R_{22}R_{11}^{-1}.
\]

Hence the primary residual vector is obtained as

\[
e = B_eZ = \Gamma_n^+(Y + \Delta Y) - \Gamma_n^+R_{22}R_{11}^{-1}U.
\]

### 2.3 Structured residual generation

Once the residual model is derived, a transformation matrix \(W\) is determined so that a given incidence matrix of the form in Table 1 is satisfied. The matrix \(W\) is acting on the primary residuals to generate the structured residuals. We first describe the design for matrix \(W\) in detail for the static case. Then we briefly mention how we extend the method to dynamic systems.

### 2.3.1 Static systems

Here we review a recent approach developed by authors of [13]. Then we present our new method.

**Structured Residual Approach with Maximized Sensitivity (SRAMS) design criterion [13]**

Given a residual model \(B_e\) and an incidence matrix in the form of Table 1, we choose \(w_i\) such that \(r_i(t)\) is insensitive to the \(i\)th sensor fault but most sensitive to the others. Mathematically, this is equivalent to

\[
\max_{j=1,\ldots,n} (w_i^T h_j^0)^2
\]

subject to the constraints in a given incidence matrix

\[
w_i^T h_j^0 = 0,
\]

with

\[
\|v\|_1 = 1, \quad h_j^0 = B_{ij}, \quad j=1,2,\ldots,n.
\]

The advantage of this earlier method is that it has a closed-form analytical solution. However, from the objective function, it can be seen that it only optimizes the sum of the squares of \(w_i^T h_j^0\). In other words, if one of the elements in the sum is large while the other elements are small, the objective function will be maximized. Referring to the incidence matrix in Table 1, this situation is highly undesirable as we want all the “non-zero” elements to be large comparing to the “zero” elements so that we can clearly differentiate faults from the residuals. There are two reasons that we want the “non-zero” elements to be large. First, due to the presence of measurement noise, disturbances, and residual modeling errors, the “zero” elements are actually non-zero, i.e. they lie in a band. Second, if the “non-zero” elements have small magnitudes that are somehow close to the magnitudes of the actual “zero” elements, we cannot isolate faults. So a better procedure should maximize the minimum of all \((w_i^T h_j^0)^2\). Thus, the smallest “nonzero” element will be far away from the “zero” elements. Hence the overall FDI system will be more robust to measurement noise, disturbances, and residual modeling errors. Our approach in the next section exactly achieves this goal.

**New Max-Min Design Criterion**

Given a residual model \(B_e\) and an incidence matrix in the form of Table 1, we choose a \(w_i\) such that the following objective function,
\[ \left\{ \min \left( w_j^T b_j^0 \right)^2, \ j = 1, \ldots, n, j \neq i \right\} \] (28)

is maximized subjecting to the constraints in a given incidence matrix

\[ w_i^T b_i^0 = 0 \]

with

\[ \|v\| = 1, \]

\[ b_j^0 = \frac{B_{ij}}{\|B_{ij}\|}, \ j = 1, 2, \ldots, n. \]

As explained earlier, the goal is to find weights \( w_0 \) such that the minimum of the nonzero elements in the incidence matrix is maximized. The maximization is done over a set of randomly selected potential vectors. To satisfy the constraint \( w_i^T b_i^0 = 0 \), the vector \( w_i \) must be chosen such that \( w_i \in S_0 \) with \( S_0 \) denoting the orthogonal vector complement space of \( b_i \). In other words,

\[ w_j = (I - b_i^0 b_i^0)^T z_j, \ z_j \in \mathbb{R}^n. \]

Note that \((I - b_i^0 b_i^0)^T\) is the orthogonal complement of \( b_i \). Denoting

\[ \left[ b_i^0 \ldots b_i^0 \right] = B_i^0 \]

and

\[ (I - b_i^0 b_i^0)^T [b_i^0 \ldots b_i^0] = (I - b_i^0 b_i^0)^T B_i^0 = B_i^0. \]

Now we form a matrix \( R = B_i^0 B_i^0^T \). There are three key observations on \( R \). First, \( R \) is symmetric. Second, using \((I - b_i^0 b_i^0)^T y_i^0 = 0\), it can be easily proven that \( Rh_i = 0 \), which means \( h_i \) is an eigenvector corresponding to the zero eigenvalues of \( R \). Third, all the eigenvectors corresponding to non-zero eigenvalues of a symmetric matrix \( R \) are orthogonal to \( h_i \), which are the eigenvectors corresponding to the zero eigenvalues of \( R \). This is a well-known result in linear algebra.

Therefore, we need to determine all the eigenvectors corresponding to the nonzero eigenvalues of \( R \). Then we form a set, which contains all the linear combination of these eigenvectors. The potential candidate of \( w_0 \) is one of the elements in the set. Finally, we search the optimal \( w_0 \) by a procedure, which is quite similar to the well-known Monte-Carlo procedure.

Algorithm to determine \( w_0 \)

**Step 1.** Define \( R = B_i^0 B_i^0^T \) with \( B_i^0 = [I - b_i^0 b_i^0^T] b_i^0, \ldots, b_i^0 = (I - b_i^0 b_i^0^T) B_i^0 \).

**Step 2.** Find all eigenvectors of \( R \) corresponding to non-zero eigenvalues of \( R \) and denote them as \( V_j \in \mathbb{R}^n, j = 1, \ldots, L \). \( L \) is the number of nonzero eigenvalues.

**Step 3.** Generate \( M \) random vectors: \( \alpha_1, \alpha_2, \ldots, \alpha_M \in \mathbb{R}^n \) where the elements inside \( \alpha_j \) follow Gaussian distribution. Then form vectors \( W_m = \frac{1}{\sqrt{L}} \alpha_m(i)V_j, m = 1, \ldots, M \). Since \( V_j \) is orthogonal to \( g \), so does each \( W_m \). Define a normalized weight vector as \( w_m = \frac{W_m}{\|W_m\|} \). Hence the constraint \( w_i^T b_i^0 = 0 \) is satisfied. Moreover, \( w_m = \frac{W_m}{\|W_m\|} \) means the constraint \( \|w_i\| = 1 \) is also satisfied.

**Step 4.** For each \( w_m \) with \( m = 1, \ldots, M \), determine \( \min \left\{ (w_m^T b_i^0)^2, \ j = 1, \ldots, n, i \neq j \right\} \) and denote the minimum values as \( P_m(w_m) \) with \( m = 1, \ldots, M \).

**Step 5.** The solution of \( w_j \) is therefore the argument of \( \max\{P_m(w_m), m = 1, \ldots, M\} \). It should be noted that the larger the \( M \) the closer to the true optimum would be for \( w_j \). Since the search is done in an off-line manner, computational complexity is not an issue.

The performance of the maximization depends on the number of random vectors, i.e. \( M \) should be relatively large. This is similar to the Monte-Carlo algorithm. If more random vectors can be used, the value of the objective function will be bigger. The beauty of this random search procedure is that it saves a lot of computations as compared to the exhaustive approach. In our simulation and experimental studies, the number \( L \) is 11 and we chose \( M = 15,800 \). The solution is close to the true optimum because the vector \( w_i \) is almost the same as the one when we chose \( M = 75,000 \). If we perform an exhaustive search and we pick 10 points in each dimension, we would need to perform a total of \( 10^L \) searches, which is extremely computationally intensive. So our search algorithm really pays off for systems with large \( L \) and saves a lot of unnecessary computations.

**Remarks.** In structured residual design, if the number of independent measurements is sufficiently large, there are usually some degrees of freedom left besides satisfying the “zero” constraints in the incidence matrix. Conventional structured residual approaches choose \( w_0 \) to be insensitive to one or several faults of interest, but it does not maximize the sensitivity to other faults. Even in the SRAMS method, it only maximizes the sum of the squares of fault responses. In a typical structured resid-
ual design, the selection of $w_i$ is not unique and somewhat arbitrary. The arbitrariness in this design does not maximize the full potential or the available information in a system to isolate faults. With the proposed Max-min method, we achieve a unique design of $w_i$ that maximizes the sensitivity to other faults, not maximizing the sensitivity of sum of the square of the fault responses, while being insensitive to the faults of interest.

2.3.2 Dynamic systems

The first step in structured residual generation is the design of an appropriate incidence matrix. For example, if our objective is to detect a single fault out of four possible faults for a static system, the incidence matrix should be:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}
\]

However, for a dynamic system with a system order of 4, we need an incidence matrix of the form:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
\] (29)

which is significantly more complicated than static systems. This is because the elements of $Y$ involve many delayed elements of $y(k)$.

With $B_i$ obtained by EIV-subspace method, the primary residual vector is:

\[
e(k) = B_i Z(k) = B_i \begin{bmatrix} z(k-n) \\ \vdots \\ z(k) \end{bmatrix},
\]

where $n$ is system order, $z(k) \in \mathbb{R}^n$, and $z(k) = y(k) + \Delta y(k)$ with $y(k)$, $\Delta y(k)$ denoting normal sensor outputs and sensor faults, respectively. Suppose our objective is to detect the $i^{th}$ failure (sensor). We can design $W$ in a way such that:

\[
r'(k) = W(i,:)e(k) = W(i,:)B_j \begin{bmatrix} z(k-n) \\ \vdots \\ z(k) \end{bmatrix}
\]

is insensitive to the $j^{th}$ element of $z(k)$.

**New Max-Min approach to design $W$ for dynamic systems: Version 1**

Mathematically, by using our new residual design approach, we choose each row vector of $W$ to achieve the following objective function:

\[
\max \left\{ \min_{j} \left[ (W(i,:)B^t(j,i))^2, \ j = 1, \ldots, (m)(n+1), \ i \neq j \right] \right\}
\]

subject to the constraints in a given incidence matrix such as (29)

\[
W(i,:)B^t(j,i,:) = B_j(:,j) = 0,
\]

with

\[
\left\| W(i,:) \right\|_1 = 1, \ \ \ B^t(j,:) = B_j(:,j)
\]

The above is a direct extension of the max-min idea described in Section 2.3.1 for the static case. In simulations and experimental studies, we observed that the above method does not generate the best results. After a very careful evaluation of results, we found out the reason. Due to the nature of the dynamic systems, we need to repeat the incidence matrix $n$ times. In the process of solving for $W$, we only maximize the minimum of the square of $W(i,:)B^t(j,i)$ for $j = 1, \ldots, (m)(n+1)$. This means the resulting signs of $W(i,:)B^t(j,i)$, $j = 1, \ldots, (m)(n+1)$, could be positive or negative. Note that:

\[
r'(k) = \sum_{j=1}^{(m)(n+1)} W(i,:)B^t(j,i) \begin{bmatrix} z(k-n) \\ \vdots \\ z(m) \end{bmatrix}^T.
\]

So when we compute the residuals, we sum all these positive and negative values to yield $r'(k)$, which will reduce the magnitude of the final value of the nonzero residuals.

To alleviate the above practical problem, we take advantage of an observation in many practical situations. The observation is that when a fault occurs, the magnitude of the fault remains constant or at least the sign of the fault remains the same for quite a number of sampling periods. This phenomenon is true when the sensors have bias, drift, and complete failures. Hence our modification is that we try to maximize the minimum of the sum of values in $rWB$, which are contributing to the current residual from the current or delayed time samples of the same sensor outputs. That is, instead of maximizing:

\[
\max \left\{ \min_{j} \left[ (W(i,:)B^t(j,i))^2, \ j = 1, \ldots, (m)(n+1) \right] \right\}
\]

we maximize:

\[
\min \left\{ \sum_{j=1}^{m} (W(i,:)B^t(j,i))^2, \ j = 1, \ldots, m \right\}
\]

Hence our modified max-min design is as follows.
New Max-Min approach to design $W$ for dynamic systems: Version 2

We choose each row vector of $W$ to achieve the following objective function,

$$
\max \left[ \min \left( \sum_{l=1}^{r} W(l,:) B_{ij}^{l}(i,j), \quad j = 1, ..., m, \ i \neq j \right) \right]^2,
$$

subject to the constraints in a given incidence matrix such as (29)

$$
W(l,:) B_{ij}^{l}(i,i+m,...,i+mn) = 0,
$$

with

$$
\left\| W(l,:) \right\| = 1, \quad B_{ij}^{l}(i,j) = B_{ij}(i,j) \left\| B_{ij}(i,j) \right\|.
$$

The optimization algorithm remains the same as that of the static system. That is, we use a procedure similar to Monte-Carlo method to randomly select some possible combinations of candidate solutions satisfying the constraints in (32) and select the best combination satisfying (31) to be the best $W$ for a given system.

### III. EXPERIMENTAL SETUP

#### 3.1 Water tank system description

Water tank system may look simple at first glance. However, its dynamics are quite complex and involve nonlinearity. As a result, many researchers (for example, Ge and Fang [5]) like to use this to demonstrate the effectiveness of their proposed ideas.

In our laboratory, we built a water tank system involving four tanks in series (Fig. 1). Being a recirculatory dynamic system, water from Tank 4 is drained into a reservoir where it is then pumped back into Tank 1. The flow rate from Tank 4 can be adjusted via a ball valve to achieve different steady-state conditions within the system. A submersible pump transfers the water from the reservoir into Tank 1. The flow rate is controlled manually with a ball valve, which will be fixed once a suitable equilibrium point is reached. The water tank system is shown in Fig. 1. It has a dimension of 14.6 cm *14 cm * 60 cm. The water level meters are pressure transducers. There are four of them with each being located near the bottom of each tank. A flowmeter was used to monitor the input flow rate. We used a data acquisition card from National Instruments to collect data.

During normal steady-state operation, the input and output flow will remain fixed. Pressure loss at tank connections will reduce water levels of subsequent tanks (e.g. the water level in Tank 2 will be less than Tank 1, but greater than that of Tank 3). Perturbations can be introduced into the system by either oscillating the flow rate or by opening one or both of the labcocks on each tank.

#### 3.2 PC-based real-time fault detection and isolation system

The SRAMS and Max-min algorithms were implemented by using ComponentWorks (a software of National Instruments) in the PC. Figure 2 shows a user interface written by using Visual Basic of our real-time sensor validation system. The upper left corner displays the sensor status information. The lower left corner shows the type and magnitude of faults that are introduced into the sensors. This part is used to emulate sensor failures. Essentially we add sensor faults to the actual sensor outputs to introduce failures. The upper right corner shows some system control functions such as saving of residual data, starting the monitoring process, instructing new users on how to use the software, etc. The lower right corner displays the time history of the computed structured residuals.

### IV. EXPERIMENTAL VERIFICATIONS

Although the system is nonlinear in nature, the residual model was generated around the equilibrium points. Hence a linear residual model was used to per-
form sensor validation. Sampling rate was 0.1. More than 2 hours of data was collected. Residual model generation procedures for dynamic systems in Section 2.2 were used. The model was a constant matrix with a dimension of 16 by 20. Incidence matrix was in the form of (29). The transformation matrix $W$ was also designed off-line, which had dimension of 4 by 16. Since the system is dynamic, version 2 of the max-min procedure was adopted to generate $W$. It took no more than 10 minutes of PC time to finish the search process (100,000 searches). Then the model parameters and $W$ were stored in the computer.

We divide our experimental studies into two parts: sensor failure isolation using real data (processed off-line), and sensor failure isolation in real-time. Both off-line and real-time tests used the same dynamic residual model and transformation matrix. The only difference is that off-line computations used our algorithm coded in MATLAB to generate results and real-time computations used our algorithm coded by another commercial software called “ComponentWorks” from National Instruments.

### 4.1 Sensor validation (off-line processing of real data)

There are four level sensors in our water tank system. Here we will first use simulations to show the advantages of our direct residual model approach. Then we will show comparative studies between our current method with an earlier approach described in [13].

All the sensor samples in this section were collected by our PC-based monitoring system. Faults were added manually through the user interface to emulate the sensor failures.

Advantage of the proposed direct approach over the indirect approach of using $\{A,B,C,D\}$

Instead of using the state-space model $(A,B,C,D)$ to generate the residual model, we directly used the input-output measurements to obtain the residual model. Our approach eliminates the numerical errors introduced during the process of obtaining the dynamic model $(A,B,C,D)$. The incidence matrix the same as (29). In other words, residual $r_i$ is zero if sensor $i$ has a fault. In the experiments, we introduced a bias with magnitude of 0.0015, which was quite small with respect to the normal sensor output magnitude (0.01 to 0.025). Bias was added to a selected sensor outputs during samples from 1500 to 4000. As can be seen from Fig. 3(b), the indirect approach of obtaining $(A,B,C,D)$ first and then generating the residual model has many false alarms. However, our direct approach shown in Fig. 3(a) had much less false alarms. Hence our Max-min method significantly reduces the effects of numerical errors during the process of model generation.

Advantage of Max-min approach over SRAMS [13]

A recent method in sensor validation was proposed in [13] using a SRAMS design criterion. The method was reviewed in Section 2. Here we compare the effectiveness of our proposed method with SRAMS.

Figure 4 shows residuals when bias with a magnitude of 0.0015 (normal sensor output magnitude is between 0.01 and 0.025) was introduced between samples 1500 to 4000. It can be seen that our approach clearly identifies the fault as $r_2$ is zero and other residuals are non-zero. However, the SRAMS method gave ambigu-
Fig. 4. Comparison of Max-min and SRAMS methods (Sensor 2 has a bias with magnitude of 0.0015).

Fig. 5. Real-time experimental results using Max-min and SRAMS approaches.
4.2 Real-time experimental results

Both Max-min and SRAMS algorithms were implemented in PC by using ComponentWorks. Different faults were introduced into the sensor outputs during normal operations. The duration of the faults lasts about 200 samples (20 second in time). At the same time, all the structured residuals were computed in real-time and were recorded. The diagnostics were done in real-time. The structured residuals were designed in a way such that \( r_i \) gives zero response when the \( i \)th sensor fails. As can be seen from Fig. 5, the Max-min approach worked perfectly in identifying all the failures in various sensors. The SRAMS method was not working well in both cases (bias, drift) except in the precision degradation case. In the bias case, the magnitude was 0.0015. For the sensor drift case, the slope of the drift was 0.00001. Note that the scale here was different from our earlier off-line simulations. Here we have scaled up all the structured residuals by a large factor of \( 10^{10} \) for ease of visualization.

V. CONCLUSIONS

In this paper, we have presented a robust approach to sensor failure isolation for both static and dynamic systems. Simulation and experimental results demonstrate the superiority of our method.

Future research includes extending the theory to linear time-varying systems.

ACKNOWLEDGEMENTS

This research was supported by NASA Dryden Flight Research Center under contract NAS9-00023.

REFERENCES


Roger Xu received the B.S. degree in electrical engineering from Jingsu Institute of Technology, Zhenjiang, Jiangsu, China in 1982 and the M.S. degree in electrical engineering from Xi’an Jiaotong University, Xi’an, Shanxi, China in 1988.
From 1988 to 1992, he was an assistant professor at Jiangsu Institute of Technology, and from 1993 to 1994, he was a visiting scholar at University of Erlangen, Germany. Since 1994 he has been working at Intelligent Automation, Inc. in Rockville, Maryland, USA as a principal research engineer. His main research areas include control theory and application, image and signal processing, and fault diagnostics and prognostics.

Chiman Kwan (S’85-M’93-SM’98) received his B.S. degree in electronics with honors from the Chinese University of Hong Kong in 1988 and M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington in 1989 and 1993, respectively.

From April 1991 to February 1994, he worked in the Beam Instrumentation Department of the SSC (Superconducting Super Collider Laboratory) in Dallas, Texas, where he was heavily involved in the modeling, simulation and design of modern digital controllers and signal processing algorithms for the beam control and synchronization system. He later joined the Automation and Robotics Research Institute in Fort Worth, where he applied intelligent control methods such as neural networks and fuzzy logic to the control of power systems, robots, and motors. Since July 1995, he has been with Intelligent Automation, Inc. in Rockville, Maryland. He has served as Principal Investigator/Program Manager for more than thirty five different projects. Currently, he is the Director of Research and Development, leading research efforts in signal/image processing, and controls. His current research areas include fault diagnostics and signal processing.