

6-1-2010

Analytic Confusion Matrix Bounds for Fault Detection and Isolation Using a Sum-of-Squared-Residuals Approach


Daniel J. Simon

Cleveland State University, d.j.simon@csuohio.edu

Donald L. Simon

NASA Glenn Research Center

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

 Part of the [Aerodynamics and Fluid Mechanics Commons](#), and the [Electrical and Computer Engineering Commons](#)

How does access to this work benefit you? Let us know!

Original Citation

Simon, D., & Simon, D. L. (2010). Analytic Confusion Matrix Bounds for Fault Detection and Isolation Using a Sum-of-Squared-Residuals Approach. *IEEE Transactions on Reliability*, 59, 2, 287-96.

Repository Citation

Simon, Daniel J. and Simon, Donald L., "Analytic Confusion Matrix Bounds for Fault Detection and Isolation Using a Sum-of-Squared-Residuals Approach" (2010). *Electrical Engineering and Computer Science Faculty Publications*. 16.

https://engagedscholarship.csuohio.edu/enece_facpub/16

This Article is brought to you for free and open access by the Electrical and Computer Engineering Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering and Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

ANALYTIC CONFUSION MATRIX BOUNDS FOR FAULT DETECTION AND ISOLATION USING A SUM-OF-SQUARED-RESIDUALS APPROACH

Dan Simon, *Cleveland State University*

Donald L. Simon, *NASA Glenn Research Center*

ABSTRACT

Given a system which can fail in 1 of n different ways, a fault detection and isolation (FDI) algorithm uses sensor data to determine which fault is the most likely to have occurred. The effectiveness of an FDI algorithm can be quantified by a confusion matrix, also called a diagnosis probability matrix, which indicates the probability that each fault is isolated given that each fault has occurred. Confusion matrices are often generated with simulation data, particularly for complex systems. In this paper, we perform FDI using sum-of-squared residuals (SSRs). We assume that the sensor residuals are s -independent and Gaussian, which gives the SSRs chi-squared distributions. We then generate analytic lower, and upper bounds on the confusion matrix elements. This approach allows for the generation of optimal sensor sets without numerical simulations. The confusion matrix bounds are verified with simulated aircraft engine data.

Index Terms—Aircraft turbofan engine, chi-squared distribution, confusion matrix, diagnosis probability matrix, fault detection and isolation.

ACRONYM

$C - MAPSS$	Commercial modular aero-propulsion system simulation
CCR	Correct classification rate
CNR	Correct no-fault rate
FDI	Fault detection and isolation
FNR	False negative rate
FPR	False positive rate
HPC	High pressure compressor
HPT	High pressure turbine
LPC	Low pressure compressor
LPT	Low pressure turbine
SSR	Sum of squared residual
TNR	True negative rate
TPR	True positive rate

NOTATION

CCR_i	CCR of fault i
CCR_{ij}	Marginal CCR of fault i relative to fault j
D_{ij}	Marginal detection rate of fault i relative to fault j
$f(x, k)$	Chi-squared pdf
$f(x, k, \lambda)$	Noncentral chi-squared pdf
$F(x, k)$	Chi-squared CDF
$F(x, k, \lambda)$	Noncentral chi-squared CDF
k	Number of sensors used to detect a fault
k_I	Cardinality of Y_i
k_{in}	Cardinality of Y_i
M_{0j}	Probability that no fault is detected given that fault j occurred
M_{i0}	Probability that fault i is isolated given that no fault occurred
M_{ij}	Probability that fault i is isolated given that fault j occurred
M'_{ij}	Marginal misclassification rate of fault i given that fault j occurred
$M_{i0,j}$	Marginal misclassification rate of fault i relative to fault j given no fault
n	Number of possible fault conditions
Nc	Core speed
P15	Bypass duct pressure
P24	LPC outlet pressure
Ps30	HPC outlet pressure
S_i	Normalized residual of the i th fault detection algorithm
T	Fault detection threshold
T24	LPC outlet temperature
T30	HPC outlet temperature
T48	HPT outlet temperature
Wf	Fuel flow
y_i	Residual of the i th sensor
Y_i	Sensors unique to algorithm i
Y_{ij}	Normalized residual of the j th sensor in Y_i
Y_c	Sensors common to two fault detection algorithms
Y_{cj}	Normalized residual of the j th sensor in Y_c

D. Simon is with the Cleveland State University, Cleveland, OH, USA (e-mail: d.j.simon@csuohio.edu).

D. L. Simon is with the NASA Glenn Research Center, Cleveland, OH, USA (email: donald.l.simon@grc.nasa.gov).

\mathcal{Y}_i	$\{Y_i, Y_c\}$
\mathcal{Y}_{ij}	Normalized residual of the j th sensor in \mathcal{Y}_i
μ_i	Mean of y_i
σ_i	Standard deviation of y_i

I. INTRODUCTION

MANY different methods of fault detection and isolation (FDI) have been proposed. Frequency domain methods include monitoring resonances [1], or modes [2]. Filter-based methods include observers [3], unknown input observers [4], Kalman filters [5], particle filters [6], sliding mode observers [7], H_∞ filters [8], and set membership filters [9]. There are also methods based on computer intelligence [10] that include fuzzy logic [11], neural networks [12], genetic algorithms [13], and expert systems [14]. Other methods include those based on Markov models [15], system identification [16], wavelets [17], Bayesian inference [18], control input manipulation [19], and the parity space approach [20]. Many other FDI methods have also been proposed [21], some of which apply to special types of systems.

The parity space approach to FDI compares the sensor residual vector to nominal user-specified fault vectors, and the closest fault vector is isolated as the most likely fault. If the sensor residual vectors are Gaussian, the parity space approach allows an analytic computation of the confusion matrix. The FDI approach that we propose is philosophically similar to the parity space approach, but instead of using fault vectors, we use sum-of-squared-residuals (SSRs) to detect and isolate a fault. Our approach is chosen because of its amenability to a new statistical method for the calculation of confusion matrix bounds.

A preliminary version of this paper was published as a technical report [22]. This paper has corrected proofs and expanded simulation results.

If sensor residuals are Gaussian, the SSRs have a chi-squared distribution [23]. This allows for the specification of SSR bounds for fault detection, which have a known false negative rate (FNR), and false positive rate (FPR). We can also compare the SSRs for each fault type to determine which fault is most likely to have occurred, and then find analytic bounds for fault isolation probabilities. Our FDI algorithm is new, but the primary contribution of this paper is to show how confusion matrix element bounds can be derived analytically. The FDI algorithm that we propose is fairly simple, but the confusion matrix analysis that we develop is novel, and its ideas may be adaptable to other FDI algorithms.

Our approach is to first specify the magnitude of each fault that we want to detect, along with a target FPR. For each fault, we then find the sensor set that gives the largest true positive rate (TPR) for the given FPR. Then we use statistical approaches to find confusion matrix bounds. The confusion matrix bounds are the outputs of this process. We cannot specify desired confusion matrix bounds ahead of time; the bounds are the s -dependent variables of the sensor selection process.

The goal of this paper is threefold. Our first goal is to present our SSR-based FDI algorithm, which we do in Section II. Our second goal is to derive confusion matrix bounds, which we do in Section III. Our third goal is to confirm the theory with

simulation results, which we do in Section IV using an aircraft turbofan engine model. Section V presents some discussion, and conclusions.

II. AN SSR-BASED FDI ALGORITHM

This section presents the background, and an overview of our proposed SSR-based FDI algorithm for a static, linear system. To perform FDI, sensor residuals are computed at each measurement time, and the SSRs are used. If the sensor residuals are Gaussian, then the SSRs have chi-squared distributions, which allows the formulation of analytic bounds on the confusion matrix elements as discussed in Sections III-A–III-C.

A. Sensor Residuals, and Chi-Squared Distributions

The residual of the i th sensor is denoted as y_i , and is a measurement of the difference between the sensor output and its nominal no-fault output. In the no-fault case, y_i has a zero expected value. In the fault case, the mean of y_i is μ_i . In either case, the standard deviation of y_i is σ_i . The mean μ_i depends on which fault occurs. But for simplicity of notation, we do not indicate that s -dependence in our notation. An SSR is given as

$$S = \sum_{i=1}^k (y_i/\sigma_i)^2. \quad (1)$$

1) No-Fault Condition: In the no-fault case, y_i has a zero expected value. If each y_i is a s -independent zero-mean Gaussian random variable, then S is a random variable with a chi-squared distribution [23]. We use the notation $f(x, k)$, and $F(x, k)$ to denote its pdf, and CDF respectively. We use a user-specified threshold T to detect a fault.

$$\begin{aligned} S \geq T &\rightarrow \text{fault detected} \\ S < T &\rightarrow \text{no fault detected.} \end{aligned}$$

Note that fault isolation is a different issue than fault detection. Detection of fault q means that $S_q \geq T_q$ for fault detection algorithm q . However, it may be that $S_i \geq T_i$ for more than one value of i . In that case, multiple faults have been detected, and a fault isolation algorithm is required to isolate the most likely fault.

The true negative rate (TNR) for fault i is the probability that $S_i < T_i$ given that there are no faults. The FPR for fault i is the probability that $S_i \geq T_i$ given that there are no faults. These probabilities are given as

$$\begin{aligned} \text{TNR}(T_i, k) &= F(T_i, k) \\ \text{FPR}(T_i, k) &= 1 - F(T_i, k). \end{aligned} \quad (2)$$

Fig. 1 illustrates TNR, and FPR for a chi-squared SSR. The TNR is the area to the left of the user-specified threshold $T = 25$, and the FPR is the area to the right of the threshold.

2) Fault Condition: If a fault occurs, then the y_i terms in (1) will not, in general, have a mean value of zero. In this case, S has a noncentral chi-squared distribution [23], and we use $f(x, k, \lambda)$, and $F(x, k, \lambda)$ to denote its pdf, and CDF, where λ is given as

$$\lambda = \sum_{i=1}^k \mu_i^2 / \sigma_i^2.$$

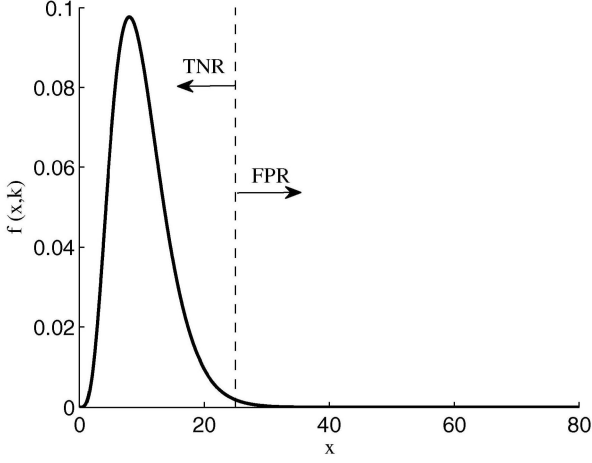


Fig. 1. Illustration of a chi-squared pdf of an SSR with $k = 10$ sensors.

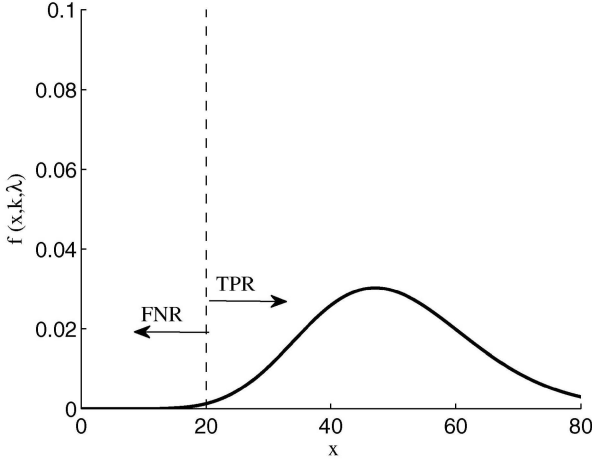


Fig. 2. Illustration of a noncentral chi-squared pdf of an SSR with $k = 10$ sensors, and $\lambda = 40$.

The TPR is defined as the probability that fault i is correctly detected ($S_i > T_i$) given that it occurs. This approach does not take fault isolation into account. The FNR is defined as the probability that fault i is not detected ($S_i < T_i$) given that it occurs. These probabilities can be written as

$$\begin{aligned} \text{FNR}_i &= F(T_i, k, \lambda) \\ \text{TPR}_i &= 1 - F(T_i, k, \lambda). \end{aligned} \quad (3)$$

Fig. 2 illustrates TPR, and FNR for a chi-squared SSR. The FNR is the area to the left of the user-specified threshold $T = 20$, and the TPR is the area to the right of the threshold.

B. Confusion Matrix

A confusion matrix specifies the likelihood of isolating each fault, and can be used to quantify the performance of an FDI algorithm. A typical confusion matrix is shown in Table I. The rows correspond to fault conditions, and the columns correspond to fault isolation results. The element in the i th row and j th column is the probability that fault j is isolated when fault i occurs. Ideally, the confusion matrix would be an identity matrix, which would indicate perfect fault isolation.

TABLE I
TYPICAL CONFUSION MATRIX FORMAT, WHERE THE ROWS CORRESPOND TO FAULT CONDITIONS, AND THE COLUMNS CORRESPOND TO FAULT ISOLATION RESULTS

	Fault 1	Fault 2	...	Fault n	No fault
Fault 1	CCR_1	M_{21}	...	M_{n1}	M_{01}
Fault 2	M_{12}	CCR_2	...	M_{n2}	M_{02}
...
Fault n	M_{1n}	M_{2n}	...	CCR_n	M_{0n}
No fault	M_{10}	M_{20}	...	M_{n0}	CNR

C. Summary of SSR-Based FDI Algorithm

Our FDI approach is to first specify the magnitude of each fault that we want to detect, along with a maximum allowable FPR. For each fault, we then find the sensor set that gives the largest TPR for the given FPR. This idea can be seen by examining Figs. 1 and 2. For a given fault, we will obtain different Figs. 1 and 2 for each possible sensor set. Given a particular Fig. 1 for a specific sensor set, we obtain a detection threshold T that corresponds to our allowable FPR. Given a particular threshold T , we obtain a TPR from Fig. 2. Intuitively we want to use sensors with large fault signatures in our FDI algorithm, and this result leads to the algorithm shown in Fig. 3 for selecting a sensor set for each fault.

Note that, although the sensor selection algorithm is logical, it is not necessarily optimal for FDI. The sensor selection algorithm in Fig. 3 is executed once for each fault that we want to detect. After we have selected a sensor set for each fault, any SSR that is greater than its threshold is considered to have been detected. If more than one SSR is greater than its threshold, the SSR that is largest relative to its threshold is isolated as the most likely fault. The FDI algorithm is summarized in Fig. 4. The strategy of isolating a fault using relative SSR values is a reasonable ad-hoc approach, but is not necessarily optimal.

III. CONFUSION MATRIX BOUNDS

This section derives analytic confusion matrix bounds for our SSR-based FDI algorithm. Section III-A deals with the no-fault case, and derives bounds for the correct no-fault rate (CNR), which is the probability that no fault is detected given that no fault occurs. It also derives bounds for the FPR, which is the probability that one or more faults are detected given that no fault occurred. Finally, it derives an upper bound for the no-fault misclassification rate, which is the probability that a given fault is isolated given that no fault occurred. Section III-B deals with the fault case, and derives bounds for the correction classification rate (CCR), which is the probability that a given fault is correctly isolated given that it occurred. Section III-C also deals with the fault case, and derives upper bounds for the fault misclassification rate, which is the probability that an incorrect fault is isolated given that some other fault occurred. Section III-D summarizes the bounds, and their use in the confusion matrix; and Section III-E discusses the required computational effort.

A. No-Fault Case

1) *Correct No-Fault Rate*: First, suppose that only two fault detection algorithms, q , and j , are running. Algorithm q at

```

Specify the maximum allowable FPR for a given fault
Initialize the sensor set to the empty set
for  $i = 1$  to number of sensors
    Add the sensor with the  $i$ th largest fault signature to the sensor set
    Use Fig. 1 to find the detection threshold corresponding to the specified FPR
    Use Fig. 2 to find the TPR corresponding to the detection threshold
next  $i$ 
Select the sensor set that has the largest TPR

```

Fig. 3. Sensor selection algorithm for a specific fault.

```

for  $j = 1$  to number of fault possibilities
    Compute the SSR  $S_j$  as shown in (1)
next  $j$ 
If  $S_j < T_j$  for all  $j$ 
    Indicate that no faults have occurred
else
    Isolated fault  $\hat{p} = \operatorname{argmax}_p (S_p - T_p)$ 
end if

```

Fig. 4. SSR-based FDI algorithm.

tempts to detect fault q using k_q sensors, and threshold T_q . We use the notation

$$\begin{aligned}
Y_i &= \{\text{sensors unique to algorithm } i\} \\
Y_c &= \{\text{sensors common to algorithms } q \text{ and } j\} \\
\mathcal{Y}_i &= \{\text{all sensors used by algorithm } i\} = \{Y_i, Y_c\} \\
k_i &= |Y_i| \\
k_{ia} &= |\mathcal{Y}_i|.
\end{aligned}$$

We use the notation \mathcal{Y}_{ij} to denote the j th normalized residual of the sensors used in algorithm i , with similar meanings for Y_{ij} , and Y_{cj} . That is,

$$S_i = \sum_j \mathcal{Y}_{ij}^2 = \sum_j Y_{ij}^2 + \sum_j Y_{cj}^2. \quad (4)$$

Now suppose that there are $n > 2$ fault detection algorithms. In this case, we can write the correct no-fault rate (CNR), which is the probability that all of the SSRs are below their detection thresholds given that no fault occurred.

$$\text{CNR} = P[(S_1 < T_1), \dots, (S_n < T_n)]. \quad (5)$$

Theorem 1: The CNR can be bounded as

$$\prod_{i=1}^n \text{TNR}(T_i, k_i) \leq \text{CNR} \leq \min_i \text{TNR}(T_i, k_i)$$

where $\text{TNR}(T_i, k_i)$ is given in (2).

Proof: See the Appendix.

2) *Fault Misclassification Rates in the No-Fault Case:* Given that no fault occurred, the probability that fault i is incorrectly isolated is called the misclassification rate, M_{i0} . In this section, we derive upper bounds for this probability.

Suppose that we have only two fault detection algorithms: q , and j . Given that no fault occurred, the probability that fault q is isolated is called the marginal misclassification of fault q relative to fault j , and is given as

$$M_{q0,j} = P[(S_q > T_q), (S_q - T_q > S_j - T_j)].$$

Lemma 1: If neither Y_q , Y_j , nor Y_c are empty, then

$$M_{q0,j} = \int_{-\infty}^{\infty} [1 - F(y, k_q)] [F(y + T_j - T_q, k_j) f(T_q - y, k_c) + f(y + T_j - T_q, k_j) (1 - F(T_q - y, k_c))] dy. \quad (6)$$

If Y_q is empty, and Y_j and Y_c are not empty, then

$$M_{q0,j} = F(T_j - T_q, k_j) [1 - F(T_q, k_c)]. \quad (7)$$

If Y_j is empty, but Y_q and Y_c are not empty, then

$$M_{q0,j} = [1 - F(T_q - T_j, k_q)] [1 - F(T_j, k_c)] + \int_{T_q - T_j}^{T_q} [1 - F(y, k_q)] f(T_q - y, k_c) dy. \quad (8)$$

If Y_c is empty, but Y_q and Y_j are not empty, then

$$M_{q0,j} = [1 - F(T_q, k_q)] F(T_j, k_j) + \int_{T_q}^{\infty} [1 - F(y, k_q)] f(y + T_j - T_q, k_j) dy. \quad (9)$$

Proof: Equation (6) can be obtained using Lemmas 5, 6, 7 and 10, which are listed in the Appendix. Equation (7) follows from the s -independence of Y_j , and Y_c . Equation (8) can be obtained using Lemmas 5, 7, and 11. Equation (9) can be obtained using Lemmas 5, 6, and 11. QED

The preceding lemma leads to the following result for the fault misclassification rate in the no-fault case.

Theorem 2: If we have $n > 2$ fault detection algorithms, the probability that fault q is isolated given that no fault occurred can be bounded as

$$M_{q0} \leq \min_{j \neq q} M_{q0,j}$$

where $M_{q0,j}$ is given by one of (6)–(9) for each j .

Proof: See the Appendix.

B. Correct Fault Classification Rates

Given that some fault occurs, we might isolate the correct fault, or we might isolate an incorrect fault. The probability of isolating the correct fault is called the correct classification rate (CCR). In this section, we derive lower, and upper bounds for the CCR.

1) *Lower Bounds for the Correct Classification Rate:* Suppose we have only two fault detection algorithms, q and j , and fault q occurs. Consider the probability that S_q is larger than S_j relative to their thresholds. We call this probability the marginal detection rate D_{qj} . Note that we are not considering whether or not the SSRs exceed their threshold; we are only considering how large the SSRs are relative to their thresholds. The marginal detection rate is given as

$$D_{qj} = P[(S_q - T_q) > (S_j - T_j)] \quad (10)$$

$$= P\left[\left(\sum_i Y_{qi}^2 - T_q + T_j\right) > \sum_i Y_{ji}^2\right]. \quad (11)$$

Lemma 2: If neither Y_q nor Y_j are empty, then

$$D_{qj} = \int_0^\infty (1 - F(y + T_q - T_j, k_q, \lambda_q)) f(y, k_j) dy \quad (12)$$

where

$$\lambda_q = \sum_i Y_{qi}^2 / \sigma_{qi}^2. \quad (13)$$

If Y_q is empty, and Y_j is not empty, then

$$D_{qj} = F(T_j - T_q, k_j). \quad (14)$$

If Y_j is empty, and Y_q is not empty, then

$$D_{qj} = 1 - F(T_q - T_j, k_q, \lambda_q). \quad (15)$$

Proof: Equation (12) can be obtained using Lemmas 5, and 6, which are in the Appendix. Equations (14), and (15) follow from (11). QED

The preceding lemma leads to the following result for the correct fault isolation rate.

Theorem 3: If we have $n > 2$ fault detection algorithms, and fault q occurs, the probability that fault q is correctly isolated is bounded as

$$CCR_q \geq TPR_q \prod_{q \neq j} D_{qj}.$$

Proof: See the Appendix.

2) *Upper Bounds for the Correction Classification Rate:* Next, we find an upper bound for the CCR. To begin, suppose that we have only two fault detectors: algorithms q , and j . Given that fault q occurs, the probability that it is correctly isolated is called the marginal CCR. This CCR can be written as

$$\begin{aligned} CCR_{qj} &= P[(S_q - T_q > S_j - T_j), (S_q > T_q)] \\ &= P\left[\sum_i Y_{qi}^2 > \max\left(\sum_i Y_{ji}^2 + T_q - T_j, T_q - \sum_i Y_{ci}^2\right)\right]. \end{aligned}$$

Lemma 3: If neither Y_q , Y_j , nor Y_c are empty, then

$$\begin{aligned} CCR_{qj} &= \int_{-\infty}^{\infty} [1 - F(y, k_q, \lambda_q)] \\ &\quad \times [F(y + T_j - T_q, k_j) f(T_q - y, k_c, \lambda_c) \\ &\quad + f(y + T_j - T_q, k_j) \\ &\quad \times (1 - F(T_q - y, k_c, \lambda_c))] dy \end{aligned} \quad (16)$$

where λ_c is defined analogously to λ_q , shown in (13). If Y_q is empty, but Y_j and Y_c are not empty, then

$$CCR_{qj} = F(T_j - T_q, k_j) [1 - F(T_q, k_c, \lambda_c)]. \quad (17)$$

If Y_j is empty, but Y_q and Y_c are not empty, then

$$\begin{aligned} CCR_{qj} &= [1 - F(T_q - T_j, k_q, \lambda_q)] [1 - F(T_j, k_c, \lambda_c)] \\ &\quad + \int_{T_q - T_j}^{T_q} [1 - F(y, k_q, \lambda_q)] f(T_q - y, k_c, \lambda_c) dy. \end{aligned} \quad (18)$$

If Y_c is empty, but Y_q and Y_j are not empty, then

$$\begin{aligned} CCR_{qj} &= [1 - F(T_q, k_q, \lambda_q)] F(T_j, k_j) \\ &\quad + \int_{T_q}^{\infty} [1 - F(y, k_q, \lambda_q)] f(y + T_j - T_q, k_j) dy. \end{aligned} \quad (19)$$

Proof: Equation (16) can be obtained using Lemmas 5, 6, 7, and 10, which are in the Appendix. Equation (17) follows from the s -independence of Y_j , and Y_c . Equation (18) can be obtained using Lemmas 5, 7, and 11. Equation (19) can be obtained using Lemmas 5, 6, and 11. QED

The preceding lemma leads to the following result for the correct fault isolation rate.

Theorem 4: If we have $n > 2$ fault detection algorithms, and fault q occurs, the probability that fault q is correctly detected and isolated can be bounded as

$$CCR_q \leq \min_{j \neq q} CCR_{qj}.$$

Proof: See the Appendix.

C. Fault Misclassification Rates

In this section, we derive upper bounds for the probability that a fault is incorrectly isolated. If fault q occurs, the probability that fault j is detected and isolated is called the misclassification rate M_{jq} .

First, suppose that we have two fault detection algorithms: q , and j . The misclassification rate can then be written as

$$\begin{aligned} M'_{jq} &= P[(S_j - T_j > S_q - T_q), (S_j > T_j)] \\ &= P\left[\sum_i Y_{ji}^2 > \max\left(\sum_i Y_{qi}^2 + T_j - T_q, T_j - \sum_i Y_{ci}^2\right)\right] \end{aligned} \quad (20)$$

where the prime symbol on M'_{jq} denotes that only two detection algorithms are used.

Lemma 4: If neither Y_q , Y_j , nor Y_c are empty, then

$$M'_{jq} = \int_{-\infty}^{\infty} [1 - F(y, k_j)] \times [F(y + T_q - T_j, k_q, \lambda_q) f(T_j - y, k_c, \lambda_c) + f(y + T_q - T_j, k_q, \lambda_q) \times (1 - F(T_j - y, k_c, \lambda_c))] dy. \quad (21)$$

If Y_q is empty, but Y_j and Y_c are not empty, then

$$M'_{jq} = [1 - F(T_j - T_q, k_j)] [1 - F(T_q, k_c, \lambda_c)] + \int_{T_j - T_q}^{T_j} [1 - F(y, k_j)] f(T_j - y, k_c, \lambda_c) dy. \quad (22)$$

If Y_j is empty, but Y_q and Y_c are not empty, then

$$M'_{jq} = F(T_q - T_j, k_q, \lambda_q) [1 - F(T_j, k_c, \lambda_c)]. \quad (23)$$

If Y_c is empty, but Y_q and Y_j are not empty, then

$$M'_{jq} = [1 - F(T_j, k_j)] F(T_q, k_q, \lambda_q) + \int_{T_j}^{\infty} [1 - F(y, k_j)] f(y + T_q - T_j, k_q, \lambda_q) dy. \quad (24)$$

Proof: Equation (21) can be obtained using Lemmas 5, 6, 7, and 10, which are in the Appendix. Equation (22) can be obtained using Lemmas 5, 7, and 11. Equation (23) follows from (20), and the s -independence of Y_q , and Y_c . Equation (24) follows from Lemmas 5, 6, and 11.

The preceding lemma leads to the following results for the fault misclassification rate.

Theorem 5: If we have $n > 2$ fault detection algorithms, and fault q occurs, the probability that fault j will be incorrectly detected and isolated can be bounded as

$$M_{jq} \leq M'_{jq}.$$

Proof: See the Appendix.

Theorem 6: The probability M_{0q} that no fault is detected when fault q occurs can be bounded from above as

$$M_{0q} \leq F(T_q, k_{qa}, \lambda_{qa}).$$

Proof: See the Appendix.

D. Summary of Confusion Matrix Bounds

Recall the confusion matrix in Table I. The rows correspond to fault conditions, and the columns correspond to fault isolation results. The element in the i th row and j th column is the probability that fault j is isolated when fault i occurs. The previous sections derived the following bounds.

- CNR is the probability that a no-fault condition is correctly indicated given that no fault occurs, and its lower, and upper bounds are given in Theorem 1.
- M_{i0} for $i \in [1, n]$ is the probability that fault i is incorrectly isolated given that no fault occurs, and its upper bound is given in Theorem 2.
- CCR_i for $i \in [1, n]$ is the probability that fault i is correctly isolated given that it occurs, and its lower, and upper bounds are given in Theorems 3 and 4.

- M_{ij} for $i, j \in [1, n]$, and $i \neq j$ is the probability that fault i is incorrectly isolated given that fault j occurs, and its upper bound is given in Theorem 5.
- M_{0i} for $i \in [1, n]$ is the probability that no fault is isolated given that fault i occurs, and its upper bound is given in Theorem 6.

E. Computational Effort

Usually, confusion matrices are obtained through simulations. To derive an experimental confusion matrix with n faults, the number of matrix elements that need to be calculated is on the order of n^2 . Also, the required number of simulations for each matrix element calculation is on the order of n . This size is because, as the number of possible faults increases, the number of simulations required to obtain the same statistical accuracy increases in direct proportion. Therefore, the computational effort required for the experimental determination of a confusion matrix is on the order of n^3 .

The bounds derived in this paper also require computational effort that is on the order of n^3 . This size is because each of the bounds summarized in Section III-D required computational effort on the order of n , and the number of matrix elements is on the order of n^2 . Note that this size does not include the sensor selection algorithm shown in Fig. 3, which requires the off-line solution of a discrete minimization problem.

IV. SIMULATION RESULTS

In this section, we use simulation results to verify the theoretical bounds of the preceding sections. We consider the problem of isolating an aircraft turbofan engine fault, which is modeled by the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [25]. There are five possible faults that can occur: fan, low pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT), and low pressure turbine (LPT). These five faults entail shifts of both efficiency, and flow capacity from nominal values. The fault magnitudes that we try to detect are 2.5% for the fan, 20% for the LPC, 2% for the HPC, 1.5% for the HPT, and 2% for the LPT. These magnitudes were chosen to give reasonable fault detection ability.

The available sensors, and their standard deviations are shown in Table II. Recall that our FDI algorithm assumes that the sensor noises are s -independent. In reality, they may have some correlation. For example, if the aircraft is operating in high humidity, all of the pressure sensors may be slightly biased in a similar fashion. However, the sensor noise correlation is a second order effect, and so we make the simplifying but standard assumption that the correlations are zero. This assumption is conceptually similar to our simplifying assumption of Gaussian noise.

The fault influence coefficient matrix shown in Table III was generated using C-MAPSS, and is based on [26]. The numbers in Table III are the partial derivatives of the sensor outputs with respect to the fault conditions, normalized to the fault percentages discussed above, and normalized to one standard deviation of the sensor noise.

We used the algorithm shown in Fig. 3 to select sensors for each fault with a maximum allowable FPR of 0.0001. As an example, consider the fan fault with the normalized fault signatures

TABLE II
AIRCRAFT ENGINE SENSORS, AND STANDARD DEVIATIONS AS A PERCENTAGE
OF THEIR NOMINAL VALUES

Symbol	Description	Standard deviation
Nc	Core speed	0.25
P15	Bypass duct pressure	0.50
P24	LPC outlet pressure	0.50
Ps30	HPC outlet pressure	0.50
T24	LPC outlet temperature	0.20
T30	HPC outlet temperature	0.17
T48	HPT outlet temperature	0.53
Wf	Fuel flow	0.75

TABLE III
FAULT SIGNATURES OF FIVE DIFFERENT FAULT CONDITIONS, WITH MEAN
SENSOR VALUE RESIDUALS NORMALIZED TO ONE STANDARD DEVIATION

	Sensors							
	Nc	P15	P24	Ps30	T24	T30	T48	Wf
Fan	0.80	2.10	1.80	4.05	0.43	0.49	1.21	3.40
LPC	0.00	0.00	4.80	1.20	3.20	0.20	0.80	0.27
HPC	0.72	0.08	0.32	0.64	0.54	5.23	3.08	1.60
HPT	0.96	0.12	0.39	3.27	0.72	2.63	4.40	2.18
LPT	1.20	0.12	0.60	3.56	0.90	3.34	0.03	2.32

TABLE IV
POTENTIAL SENSOR SETS FOR DETECTING A FAN FAULT

Sensors	Detection threshold T_1	TPR ₁
Ps30	15.1	0.563
Ps30, Wf	18.4	0.865
Ps30, Wf, T30	21.1	0.926
Ps30, Wf, T30, P15	23.5	0.949
Ps30, Wf, T30, P15, P24	25.7	0.959
Ps30, Wf, T30, P15, P24, T48	27.9	0.958
Ps30, Wf, T30, P15, P24, T48, Nc	29.9	0.952
Ps30, Wf, T30, P15, P24, T48, Nc, T24	31.8	0.942

shown in Table III. The sensors with the largest fault signatures in descending order are Ps30, Wf, T30, P15, P24, T48, Nc, and T24. This gives eight potential sensor sets for detecting a fan fault: the first potential set uses only sensor Ps30, the second potential set uses Ps30 and Wf, and so on. The potential sensor sets, along with their detection thresholds, and TPRs, are shown in Table IV. Table IV shows that using five sensors gives the largest TPR given the constraint that $FPR \leq 0.0001$. The thresholds were determined by constraining $FPR \leq 0.0001$. Using five sensors gives the largest TPR subject to the FPR constraint.

This process described in the previous paragraph was repeated for each fault shown in Table III. The resulting sensor sets are shown in Table V. Note that, given a FPR constraint, the detection threshold is a function only of the number of sensors in each sensor set; the detection threshold is not a function of the specific fault signatures. This result is illustrated in Fig. 1, where it is seen that $f(x, k)$ is a function only of x , and k (the number of sensors).

We used the fault isolation method shown in Fig. 4, along with the theorems in the previous sections to obtain lower, and upper bounds for the confusion matrix as summarized in Section III-D. We also ran 100,000 simulations to obtain an experimental confusion matrix. Table VI shows the theoretical lower bounds of the diagonal elements of the confusion matrix. Lower bounds of the off-diagonal elements were not obtained because we are typically more interested in upper bounds of off-diagonal

TABLE V
SENSOR SETS FOR FAULT DETECTION GIVING THE LARGEST TPR FOR EACH
FAULT GIVEN THE CONSTRAINT THAT $FPR \leq 0.0001$

Fault	Sensor set	Detection threshold T	TPR
Fan	Ps30, Wf, T30, P15, P24	25.7	0.959
LPC	P24, T24	18.4	0.943
HPC	T30, T48	18.4	0.970
HPT	T48, Ps30, T30, Wf	23.5	0.970
LPT	Ps30, T30, Wf	21.1	0.844

TABLE VI
LOWER BOUNDS OF DIAGONAL CONFUSION MATRIX ELEMENTS WHERE
ROWS SPECIFY THE ACTUAL FAULT CONDITION, AND COLUMNS SPECIFY
THE DIAGNOSIS

	Fan	LPC	HPC	HPT	LPT	No Fault
Fan	0.6691					
LPC		0.9342				
HPC			0.8692			
HPT				0.9345		
LPT					0.6623	
No Fault						0.9992

TABLE VII
UPPER BOUNDS OF THE CONFUSION MATRIX ELEMENTS WHERE ROWS SPECIFY
THE ACTUAL FAULT CONDITION, AND COLUMNS SPECIFY THE DIAGNOSIS

	Fan	LPC	HPC	HPT	LPT	No Fault
Fan	0.7761	0.0000	0.0001	0.1115	0.1899	0.0408
LPC	0.0076	0.9356	0.0000	0.0000	0.0000	0.0573
HPC	0.0097	0.0000	0.8936	0.0769	0.0160	0.0303
HPT	0.0015	0.0000	0.0270	0.9445	0.0014	0.0300
LPT	0.0874	0.0000	0.0030	0.1066	0.7422	0.1557
No Fault	0.0000	0.0000	0.0000	0.0000	0.0001	0.9999

TABLE VIII
EXPERIMENTAL CONFUSION MATRIX USING SSR-BASED DI WHERE ROWS
SPECIFY THE ACTUAL FAULT CONDITION, AND COLUMNS SPECIFY THE
DIAGNOSIS, BASED ON 100,000 SIMULATIONS OF EACH FAULT

	Fan	LPC	HPC	HPT	LPT	No Fault
Fan	0.7614	0.0000	0.0000	0.0370	0.1677	0.0338
LPC	0.0073	0.9354	0.0000	0.0000	0.0000	0.0573
HPC	0.0051	0.0000	0.8875	0.0713	0.0074	0.0288
HPT	0.0016	0.0000	0.0276	0.9422	0.0011	0.0275
LPT	0.0831	0.0000	0.0015	0.0993	0.6680	0.1481
No Fault	0.0000	0.0000	0.0000	0.0000	0.0001	0.9997

elements. Table VII shows the theoretical upper bounds of the confusion matrix. Table VIII shows the experimental confusion matrix. These tables show that the theoretical results derived in this paper give reasonably tight bounds to the experimental confusion matrix values.

Recall that we used a FPR of 0.0001 to choose our sensor sets, and detection thresholds. Therefore, the first five elements in the last row of Table VII are guaranteed to be no greater than 0.0001. Further, the element in the lower right corner of Table VI is guaranteed to be no greater than $1 - 5(0.0001) = 0.9995$.

Note that it is possible for an element in the experimental confusion matrix in Table VIII to lie outside the bounds shown in Tables VI and VII (for example, see the numbers in the fourth row, and first column in Tables VII and VIII). This result is true because the numbers in Table VIII are experimentally obtained on the basis of a finite number of simulations, and are guaranteed to lie within their theoretical bounds only as the number of simulations approaches infinity. In fact, that is one of the strengths of

TABLE IX
EXPERIMENTAL CONFUSION MATRIX USING THE PARITY-SPACE APPROACH
FOR FDI, BASED ON 100,000 SIMULATIONS OF EACH FAULT

	Fan	LPC	HPC	HPT	LPT	No Fault
Fan	0.9421	0.0001	0.0000	0.0000	0.0000	0.0579
LPC	0.0002	0.8210	0.0000	0.0000	0.0000	0.1789
HPC	0.0000	0.0000	0.9072	0.0000	0.0006	0.0922
HPT	0.0000	0.0000	0.0000	0.9360	0.0000	0.0640
LPT	0.0000	0.0000	0.0014	0.0000	0.7301	0.2685
No Fault	0.0000	0.0000	0.0000	0.0000	0.0000	0.9999

the analytic method proposed in this paper. The analytic bounds are definite, but simulations are subject to random effects. Also, simulations can give misleading conclusions if the simulation has errors. One common simulation error is the non-randomness of commonly used pseudorandom number generators [27].

To summarize the SSR-based FDI algorithm, the user specifies the maximum FPR for each fault, and then finds the sensor set that has the largest TPR given the FPR constraint. Analytic confusion matrix bounds are then obtained using the theory in this paper. If the results are not satisfactory, the user can iterate by changing the maximum FPR constraint. For example, if a TPR is too small, then the user will have to increase the FPR constraint. If the confusion matrix bounds of fault isolation probabilities are not satisfactory, the user will have to iterate on the FPR constraints to obtain different confusion matrix bounds.

We also generated FDI results using the parity space approach [20] to explore the relative performance of our new SSR-based FDI approach. The parity space approach uses all sensors for all fault detectors, and we set the detection thresholds to achieve an FPR of 0.0001 to be consistent with the SSR-based approach. Results are shown in Table IX. A comparison of Tables VIII and IX shows that the parity space approach generally performs better than the SSR-based approach, although the results are comparable. The confusion matrix in Table VIII for the SSR-based algorithm has a condition number of 1.83, while the matrix in Table IX for the parity space approach has a condition number of 1.65. This result shows that the confusion matrix for the parity space approach is about 9.8% closer to perfect than the confusion matrix for the SSR-based approach.

V. CONCLUSION

This paper has introduced a new FDI algorithm, and derived analytical confusion matrix bounds. The main contribution of this paper is the generation of analytic confusion matrix bounds, and the possibility that our methodology could be adapted to other FDI algorithms. Usually, confusion matrices are obtained with simulations. Such simulations have several potential drawbacks. First, they can be time consuming. Second, they can give misleading conclusions if not enough simulations are run to give statistically significant results. Third, they can give misleading conclusions if the simulation has errors (for example, if the output of the random number generator does not satisfy statistical tests for randomness). The theoretical confusion matrix bounds derived in this paper do not depend on a random number generator, and can be used in place of simulations.

Further work in this area could follow several directions. First, the tightness of the confusion matrix bounds could be

quantified. This paper derives bounds, but does not guarantee how loose or tight those bounds are. Second, the bounds could be modified to be tighter. Third, bounds could be attempted for methods other than the FDI algorithm proposed here. The fault isolation method we used isolates the fault that has the largest SSR relative to its detection threshold. Other fault isolation methods could normalize the relative SSR to its standard deviation, or could normalize the SSR to its detection threshold. Our FDI method is static, which means that faults are isolated using measurements at a single time. Better fault isolation might be achieved if dynamic system information is used.

APPENDIX

We use the following lemmas to derive the results of this paper. We use the notation $f_x(w)$, and $F_x(w)$ to denote the pdf, and CDF of the random variable x evaluated at w . If the random variable is clear from the context, we shorten the notation to $f(x)$, and $F(x)$ respectively. These lemmas can be proven using standard definitions, and results from probability theory [24].

Lemma 5: The probability that a realization of the random variable x is greater than a realization of the random variable y is given as

$$P(x > y) = \int_{-\infty}^{\infty} \int_y^{\infty} f(x, y) dx dy$$

where $f(x, y)$ is the joint pdf of x , and y . If x , and y are s -independent, this result can be written as

$$\begin{aligned} P(x > y) &= \int_{-\infty}^{\infty} \left[\int_z^{\infty} f_x(w) dw \right] f_y(z) dz \\ &= \int_{-\infty}^{\infty} [1 - F_x(z)] f_y(z) dz. \end{aligned}$$

Lemma 6: If $y = T + x$, where x is a random variable, and T is a constant, then

$$\begin{aligned} f_y(w) &= f_x(w - T) \\ F_y(w) &= F_x(w - T). \end{aligned}$$

Lemma 7: If $y = T - x$, where x is a random variable, and T is a constant, then

$$\begin{aligned} f_y(w) &= f_x(T - w) \\ F_y(w) &= 1 - F_x(T - w). \end{aligned}$$

Lemma 8: If $z = \min(x, y)$, where x and y are s -independent random variables, then

$$f_z(w) = f_x(w) (1 - F_y(w)) + f_y(w) (1 - F_x(w)).$$

Lemma 9: If $z = \min(x, T)$, where x is a random variable, and T is a constant, then

$$\begin{aligned} F_z(w) &= \begin{cases} F_x(w) & w < T \\ 1 & w \geq T \end{cases} \\ f_z(w) &= \begin{cases} f_x(w) & w < T \\ 0 & w > T \\ (1 - F_x(w)) \delta(w - T) & w = T \end{cases} \end{aligned}$$

where $\delta(\cdot)$ is the continuous-time impulse function.

Lemma 10: If $z = \max(x, y)$, where x and y are s -independent random variables, then

$$f_z(w) = f_x(w)F_y(w) + f_y(w)F_x(w).$$

Lemma 11: If $z = \max(x, T)$, where x is a random variable, and T is a constant, then

$$F_z(w) = \begin{cases} F_x(w) & w \geq T \\ 0 & w < T \end{cases}$$

$$f_z(w) = \begin{cases} f_x(w) & w > T \\ 0 & w < T \\ F_x(w)\delta(w - T) & w = T. \end{cases}$$

Proof of Theorem 1: Equation (5) gives the definition of CNR as

$$CNR = P[(S_1 < T_1), \dots, (S_n < T_n)]$$

where the T_i are constant, and the S_i are random variables. If none of the fault detection algorithms have any sensors in common, then each S_i is s -independent, which means that

$$CNR = P(S_1 < T_1) \cdots P(S_n < T_n)$$

$$= \prod_{i=1}^n \text{TNR}(T_i, k_i).$$

If the algorithms have common sensors, then the S_i terms are positively s -dependent, which will increase the CNR. On the other hand, if there is some m such that \mathcal{Y}_m is a superset of \mathcal{Y}_i for all $i \neq m$, $\text{TNR}(T_m, k_m) \leq \text{TNR}(T_i, k_i)$ for all $i \neq m$, and $T_m \leq T_i$ for all $i \neq m$, then $S_m < T_m \implies S_i < T_i$ for all i , which means that

$$CNR = \text{TNR}(T_m, k_m).$$

QED

Proof of Theorem 2: Given $n > 2$ fault detection algorithms, the probability that fault q is isolated given that no fault occurred is the probability that S_q is greater than its threshold, and also greater than all of the other SSRs relative to their thresholds.

$$M_{q0} = P[(S_q > T_q), (S_q - T_q > S_j - T_j \text{ for all } j \neq q)]$$

$$\leq \min_{j \neq q} P[(S_q > T_q), (S_q - T_q > S_j - T_j)]$$

$$\leq \min_{j \neq q} M_{q0,j}.$$

QED

Proof of Theorem 3: First, we establish the positive s -dependence [28, p. 145] of the random variables $S_q - S_j$ for all $j \neq q$. Consider inequalities $S_q - S_j > T_q - T_j$ for $j \neq q$. It follows from (4) that $S_q - S_j$ is an increasing function of the negative squared normalized residuals of the common sensors of S_j , and thus the random variables $S_q - S_j, j \neq q$, are positively dependent.

Now note that, if fault q occurred, then the probability that S_q is larger than S_j relative to its threshold for all $j \neq q$ is given as

$$D_q = P(S_q - T_q > S_j - T_j \text{ for all } j \neq q) \geq \prod_{j \neq q} D_{qj}$$

where the inequality comes from the positive dependence of $S_q - S_j, j \neq q$. The probability that fault q is isolated given that fault q occurred can be written as

$$\text{CCR}_q = P[(S_q > T_q), (S_q - T_q > S_j - T_j \text{ for all } j \neq q)]$$

$$\geq \text{TPR}_q D_q$$

where the inequality comes from the positive dependence of the random variables S_q , and $S_q - S_j, j \neq q$. QED

Proof of Theorem 4: If we have $n > 2$ fault detection algorithms, and fault q occurs, the probability that fault q is correctly detected and isolated is the probability that S_q is greater than its threshold, and also greater than all other SSRs relative to their thresholds.

$$\text{CCR}_q = P[(S_q > T_q), (S_q - T_q > S_j - T_j \text{ for all } j \neq q)]$$

$$\leq \min_{j \neq q} P[(S_q > T_q), (S_q - T_q > S_j - T_j)]$$

$$\leq \min_{j \neq q} \text{CCR}_{qj}.$$

QED

Proof of Theorem 5: Given that we have $n > 2$ fault detection algorithms, the misclassification rate M_{jq} is bounded from above by M'_{jq} . So to obtain an upper bound for M_{jq} , we use one of (21)–(24) as appropriate. This approach gives

$$M_{jq} = P[(S_j > T_j), (S_j - T_j > S_i - T_i \text{ for all } i \neq j)]$$

$$\leq P[(S_j > T_j), (S_j - T_j > S_q - T_q)]$$

$$\leq M'_{qj}.$$

QED

Proof of Theorem 6: The probability M_{0q} that no fault is detected when fault q occurs is given as

$$M_{0q} = P(S_j < T_j \text{ for all } j) \leq P(S_q < T_q) \leq F(T_q, k_{qa}, \lambda_{qa}).$$

QED

REFERENCES

- [1] S. Chinchalkar, "Determination of crack location in beams using natural frequencies," *Journal of Sound and Vibration*, vol. 247, pp. 417–429, Oct. 2001.
- [2] T. Tsai and Y. Wang, "Vibration analysis and diagnosis of a cracked shaft," *Journal of Sound and Vibration*, vol. 192, pp. 607–620, May 1996.
- [3] H. Wang, Z. Huang, and D. Steven, "On the use of adaptive updating rules for actuator and sensor fault diagnosis," *Automatica*, vol. 33, pp. 217–224, Feb. 1997.
- [4] J. Chen, R. Patton, and H. Zhang, "Design of unknown input observers and robust fault detection filters," *International Journal of Control*, vol. 63, pp. 85–105, Jan. 1996.
- [5] J. Korbicz, J. Koscielny, Z. Kowalczyk, and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Springer, 2004.
- [6] P. Li and V. Kadiramanathan, "Fault detection and isolation in nonlinear stochastic systems—A combined adaptive Monte Carlo filtering and likelihood ratio approach," *International Journal of Control*, vol. 77, pp. 1101–1114, Dec. 2004.
- [7] C. Tan and C. Edwards, "Sliding mode observers for robust detection and reconstruction of actuator sensor faults," *International Journal of Robust and Nonlinear Control*, vol. 13, pp. 443–463, Apr. 2003.
- [8] I. Yaesh and U. Shaked, "Robust H_∞ deconvolution and its application to fault detection," *Journal of Guidance, Control and Dynamics*, vol. 23, pp. 1101–1112, Jun. 2000.
- [9] C. Ocampo-Martinez, S. Tornil, and V. Puig, "Robust fault detection using interval constraints satisfaction and set computations," in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug. 30–Sep. 1 2006, pp. 1285–1290.

- [10] W. Fenton, T. MicGinnity, and L. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: A review," *IEEE Trans. Systems, Man and Cybernetics: Part C – Applications and Reviews*, vol. 31, pp. 269–281, Aug. 2001.
- [11] H. Schneider and P. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy-logic residual evaluation," *IEEE Trans. Control System Technology*, vol. 4, pp. 274–282, May 1996.
- [12] M. Napolitano, C. Neppach, V. Casdorph, S. Naylor, M. Innocenti, and G. Silvestri, "Neural-network-based scheme for sensor failure detection, identification and accommodation," *Journal of Guidance, Control and Dynamics*, vol. 18, pp. 1280–1286, Nov. 1995.
- [13] Z. Yangping, Z. Bingquan, and W. DongXin, "Application of genetic algorithms to fault diagnosis in nuclear power plants," *Reliability Engineering and System Safety*, vol. 67, pp. 153–160, Feb. 2000.
- [14] W. Gui, C. Yang, J. Teng, and W. Yu, "Intelligent fault diagnosis in lead-zinc smelting process," in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug. 30–Sep. 1 2006, pp. 234–239.
- [15] S. Lu and B. Huang, "Condition monitoring of model predictive control systems using Markov models," in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug. 30–Sep. 1 2006, pp. 264–269.
- [16] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods—An introduction," *Control Engineering Practice*, vol. 5, pp. 639–652, May 1997.
- [17] X. Deng and X. Tian, "Multivariate statistical process monitoring using multi-scale kernel principal component analysis," in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug. 30–Sep. 1 2006, pp. 108–113.
- [18] A. Pernestal, M. Nyberg, and B. Wahlberg, "A Bayesian approach to fault isolation—Structure estimation and inference," in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug. 30–Sep. 1 2006, pp. 450–455.
- [19] S. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. : Princeton University Press, 2004.
- [20] F. Gustafsson, "Statistical signal processing approaches to fault detection," *Annual Reviews in Control*, vol. 31, pp. 41–54, Apr. 2007.
- [21] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. : CRC, 1998.
- [22] D. Simon and D. L. Simon, "Analytic confusion matrix bounds for fault detection and isolation using a sum-of-squared-residuals approach, NASA Technical Memorandum TM-2009-215655," Jul. 2009.
- [23] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*. : Dover, 1965.
- [24] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*. : McGraw-Hill, 2002.
- [25] D. Frederick, J. DeCastro, and J. Litt, User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) NASA Technical Memorandum TM-2007-215026.
- [26] D. L. Simon, J. Bird, C. Davison, A. Volponi, and R. Iverson, "Benchmarking gas path diagnostic methods: A public approach," presented at the ASME Turbo Expo, Jun. 2008, Paper GT2008-51360, unpublished.
- [27] P. Savicky and M. Robnik-Sikonja, "Learning random numbers: A Matlab anomaly," *Applied Artificial Intelligence*, vol. 22, pp. 254–265, Mar. 2008.
- [28] C. Lai and M. Xie, "Concepts of stochastic dependence in reliability analysis," in *Handbook of Reliability Engineering*, H. Pham, Ed. : Springer, 2003, pp. 141–156.

Dan Simon (S'89–M'90–SM'01) received a B.S. degree from Arizona State University (1982), an M.S. degree from the University of Washington (1987), and a Ph.D. degree from Syracuse University (1991), all in electrical engineering. He worked in industry for 14 years at Boeing, TRW, and several small companies. His industrial experience includes work in the aerospace, automotive, agricultural, GPS, biomedical, process control, and software fields. In 1999, he moved from industry to academia, where he is now a professor in the Electrical and Computer Engineering Department at Cleveland State University. His teaching and research involves embedded systems, control systems, and computer intelligence. He has published about 80 refereed conference and journal papers, and is the author of the text *Optimal State Estimation* (John Wiley & Sons, 2006).