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: [http://engagedscholarship.csuohio.edu/enece_facpub](http://engagedscholarship.csuohio.edu/enece_facpub)

Part of the *Aerodynamics and Fluid Mechanics Commons*

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

**Publisher's Statement**

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

**Original Citation**


**Repository Citation**

[http://engagedscholarship.csuohio.edu/enece_facpub/16](http://engagedscholarship.csuohio.edu/enece_facpub/16)

This Article is brought to you for free and open access by the Electrical Engineering & Computer Science Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering & 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

<table>
<thead>
<tr>
<th>ACronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C - MAPSS</td>
<td>Commercial modular aero-propulsion system simulation</td>
</tr>
<tr>
<td>CCR</td>
<td>Correct classification rate</td>
</tr>
<tr>
<td>CNR</td>
<td>Correct no-fault rate</td>
</tr>
<tr>
<td>FDI</td>
<td>Fault detection and isolation</td>
</tr>
<tr>
<td>FNR</td>
<td>False negative rate</td>
</tr>
<tr>
<td>FPR</td>
<td>False positive rate</td>
</tr>
<tr>
<td>HPC</td>
<td>High pressure compressor</td>
</tr>
<tr>
<td>HPT</td>
<td>High pressure turbine</td>
</tr>
<tr>
<td>LPC</td>
<td>Low pressure compressor</td>
</tr>
<tr>
<td>LPT</td>
<td>Low pressure turbine</td>
</tr>
<tr>
<td>SSR</td>
<td>Sum of squared residual</td>
</tr>
<tr>
<td>TNR</td>
<td>True negative rate</td>
</tr>
<tr>
<td>TPR</td>
<td>True positive rate</td>
</tr>
<tr>
<td>CCR(_i)</td>
<td>CCR of fault (_i)</td>
</tr>
<tr>
<td>CCR(_ij)</td>
<td>Marginal CCR of fault (_i) relative to fault (_j)</td>
</tr>
<tr>
<td>(D_{ij})</td>
<td>Marginal detection rate of fault (_i) relative to fault (_j)</td>
</tr>
<tr>
<td>(f(x,k))</td>
<td>Chi-squared pdf</td>
</tr>
<tr>
<td>(f(x,k,\lambda))</td>
<td>Noncentral chi-squared pdf</td>
</tr>
<tr>
<td>(F(x,k))</td>
<td>Chi-squared CDF</td>
</tr>
<tr>
<td>(F(x,k,\lambda))</td>
<td>Noncentral chi-squared CDF</td>
</tr>
<tr>
<td>(k)</td>
<td>Number of sensors used to detect a fault</td>
</tr>
<tr>
<td>(k_i)</td>
<td>Cardinality of (Y_i)</td>
</tr>
<tr>
<td>(k_{ia})</td>
<td>Cardinality of (Y_i^a)</td>
</tr>
<tr>
<td>(M_{0j})</td>
<td>Probability that no fault is detected given that fault (_j) occurred</td>
</tr>
<tr>
<td>(M_{ia})</td>
<td>Probability that fault (_i) is isolated given that no fault occurred</td>
</tr>
<tr>
<td>(M_{ij})</td>
<td>Probability that fault (_i) is isolated given that fault (_j) occurred</td>
</tr>
<tr>
<td>(M'_{ij})</td>
<td>Marginal misclassification rate of fault (_i) given that fault (_j) occurred</td>
</tr>
<tr>
<td>(M_{ii,j})</td>
<td>Marginal misclassification rate of fault (_i) relative to fault (_j) given no fault</td>
</tr>
<tr>
<td>(n)</td>
<td>Number of possible fault conditions</td>
</tr>
<tr>
<td>(Nc)</td>
<td>Core speed</td>
</tr>
<tr>
<td>P15</td>
<td>Bypass duct pressure</td>
</tr>
<tr>
<td>P24</td>
<td>LPC outlet pressure</td>
</tr>
<tr>
<td>Ps30</td>
<td>HPC outlet pressure</td>
</tr>
<tr>
<td>(S_i)</td>
<td>Normalized residual of the (_i)th fault detection algorithm</td>
</tr>
<tr>
<td>(T)</td>
<td>Fault detection threshold</td>
</tr>
<tr>
<td>T24</td>
<td>LPC outlet temperature</td>
</tr>
<tr>
<td>T30</td>
<td>HPC outlet temperature</td>
</tr>
<tr>
<td>T48</td>
<td>HPT outlet temperature</td>
</tr>
<tr>
<td>Wf</td>
<td>Fuel flow</td>
</tr>
<tr>
<td>(y_i)</td>
<td>Residual of the (_i)th sensor</td>
</tr>
<tr>
<td>(Y_i)</td>
<td>Sensors unique to algorithm (_i)</td>
</tr>
<tr>
<td>(Y_{ij})</td>
<td>Normalized residual of the (_j)th sensor in (Y_i)</td>
</tr>
<tr>
<td>(Y_c)</td>
<td>Sensors common to two fault detection algorithms</td>
</tr>
<tr>
<td>(Y_{cj})</td>
<td>Normalized residual of the (_j)th sensor in (Y_c)</td>
</tr>
</tbody>
</table>
\[ Y_i = \{ Y_i, \tilde{Y}_i \} \]
\[ Y_{ij} \] Normalized residual of the \( j \)th sensor in \( Y_i \)
\[ \mu_i \] Mean of \( y_i \)
\[ \sigma_i \] 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 [5], unknown input observers [4], Kalman filters [5], particle filters [6], sliding mode observers [7], II_{\infty} 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. 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 elements 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 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 \( \mu_i \). In either case, the standard deviation of \( y_i \) is \( \sigma_i \). The mean \( \mu_i \) 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

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

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:

\[
S \geq T \rightarrow \text{fault detected}
\]
\[
S < T \rightarrow \text{no fault detected}.
\]

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

\[
\text{TNR}(T_i, k) = F(T_i, k)
\]
\[
\text{FPR}(T_i, k) = 1 - F(T_i, k).
\]

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 \( \lambda \) is given as

\[
\lambda = \sum_{i=1}^{k} \rho_i^2/\sigma_i^2.
\]
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

$$\text{FNR}_i = F(T_i, k, \lambda)$$

$$\text{TPR}_i = 1 - F(T_i, k, \lambda).$$

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.

**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 misclassification 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 \( \hat{p} = \text{argmax}_p (S_p - T_p) \).
end if

Fig. 4. SSR-based FDI algorithm.

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

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

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

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

If \( Y_q \) is empty, and \( Y_j \) and \( Y_c \) are not empty, then

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

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

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

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

\[
M_{q0,j} = \int_{-\infty}^{T_q} \left[ 1 - F(y, k_q) \right] f(T_q - y, k_c) dy.
\]

Proof: Equation (6) can be obtained using Lemmas 5, 6, 7, and 10, which are listed in the Appendix. Equation (7) follows from the \( x \)-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, 9, 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

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

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, and fault occurs. Consider the probability that is larger than relative to their thresholds. We call this probability the marginal detection rate . 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

\[
D_{qj} = P \left[ (S_q - T_q) > (S_j - T_j) \right] = P \left[ \left( \sum_i Y_{q_i}^2 - T_q + T_j \right) > \sum_i Y_{j_i}^2 \right].
\]

(10)

Lemma 2: If neither nor is empty, then

\[
D_{qj} = \int_0^\infty \left( 1 - F(y + T_q - T_j, k_q, \lambda_q) \right) f(y, k_j) \, dy.
\]

(12)

where

\[
\lambda_q = \sum_i Y_{q_i}^2 / \sigma_{q_i}^2.
\]

(13)

If is empty, and is not empty, then

\[
D_{qj} = F(T_j - T_q, k_j).
\]

(14)

If is empty, and is not empty, then

\[
D_{qj} = 1 - F(T_q - T_j, k_q, \lambda_q).
\]

(15)

Proof: Equation (12) can be obtained using Lemmas 5, and 6, which are in the Appendix. Equations (14), and (15) follow from (11). The preceding lemma leads to the following result for the correct fault isolation rate.

Theorem 3: If we have two fault detection algorithms, and fault occurs, the probability that fault is correctly isolated is bounded as

\[
\text{CCR}_q \geq \text{TPR}_q \prod_{q \neq q} D_{qj}.
\]

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 and . Given that fault occurs, the probability that it is correctly isolated is called the marginal CCR. This CCR can be written as

\[
\text{CCR}_{qj} = P \left[ (S_q - T_q) > (S_j - T_j), (S_q > T_q) \right]
\]

\[
= P \left[ \sum_i Y_{q_i}^2 > \max \left( \sum_i Y_{j_i}^2 + T_q - T_j, T_q - \sum_i Y_{c_i}^2 \right) \right].
\]

(20)

where the prime symbol on indicates that only two detection algorithms are used.

Lemma 3: If neither nor are empty, then

\[
\text{CCR}_{qj} = \int_{-\infty}^\infty \left[ 1 - F(y, k_q, \lambda_q) \right] \times \left[ F(y + T_q - T_j, k_j) f(T_q - y, k_c) + f(y + T_j - T_q, k_j) \times (1 - F(T_q - y, k_q, \lambda_q)) \right] \, dy.
\]

(16)

where is defined analogously to , shown in (13). If is empty, but and are not empty, then

\[
\text{CCR}_{qj} = F(T_j - T_q, k_j) \left[ 1 - F(T_q, k_c, \lambda_c) \right].
\]

(17)

If is empty, but and are not empty, then

\[
\text{CCR}_{qj} = [1 - F(T_q - T_j, k_q, \lambda_q)] \times [1 - F(T_j, k_c, \lambda_c)]
\]

\[
+ \int_{T_q - T_j}^{T_q} \left[ 1 - F(y, k_q, \lambda_q) \right] f(T_q - y, k_c, \lambda_c) \, dy.
\]

(18)

If is empty, but and are not empty, then

\[
\text{CCR}_{qj} = [1 - F(T_q, k_q, \lambda_q)] f(T_j, k_j)
\]

\[
+ \int_{T_q}^{\infty} \left[ 1 - F(y, k_q, \lambda_q) \right] f(y + T_j - T_q, k_j) \, dy.
\]

(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 , and . Equation (18) can be obtained using Lemmas 5, 7, and 11. Equation (19) can be obtained using Lemmas 5, 6, and 11.

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

Theorem 4: If we have two fault detection algorithms, and fault occurs, the probability that fault is correctly detected and isolated is bounded as

\[
\text{CCR}_q \leq \min_{j \neq q} \text{CCR}_{qj}.
\]

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 occurs, the probability that fault is detected and isolated is called the misclassification rate .

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

\[
M_{qj} = P \left[ (S_q - T_q) > (S_j - T_q), (S_j > T_j) \right]
\]

\[
= P \left[ \sum_i Y_{q_i}^2 > \max \left( \sum_i Y_{j_i}^2 + T_q - T_j, T_j - \sum_i Y_{c_i}^2 \right) \right].
\]

(20)
Lemma 4: If neither \( Y_q, Y_j \), nor \( Y_c \) are empty, then
\[
M'_{jq} = \int_{-\infty}^{\infty} \left[ 1 - F(y, k_j) \right] \\
\times \left[ F(y + T_q - T_j, k_q, \lambda_q) f(T_j - y, k_c, \lambda_c) \\
+ f(y + T_q - T_j, k_q, \lambda_q) \right] \\
\times \left[ 1 - F(T_j - y, k_c, \lambda_c) \right] dy. 
\]  
(21)

If \( Y_q \) is empty, but \( Y_j \) and \( Y_c \) are not empty, then
\[
M'_{jq} = \int_{T_j - T_q}^{\infty} \left[ 1 - F(y, k_j) \right] \\
\times \left[ F(T_j - y, k_q, \lambda_q) \right] \\
\times \left[ 1 - F(T_j - y, k_c, \lambda_c) \right] dy. 
\]  
(22)

If \( Y_j \) is empty, but \( Y_q \) and \( Y_c \) are not empty, then
\[
M'_{jq} = F(T_j - T_q, k_q, \lambda_q) \\
\times \left[ 1 - F(T_j, k_c, \lambda_c) \right]. 
\]  
(23)

If \( Y_c \) is empty, but \( Y_q \) and \( Y_j \) are not empty, then
\[
M'_{jq} = \int_{-\infty}^{\infty} \left[ 1 - F(y, k_j) \right] \\
\times \left[ F(y + T_q - T_j, k_q, \lambda_q) \right] \\
\times \left[ 1 - F(y + T_q - T_j, k_c, \lambda_c) \right] dy. 
\]  
(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
\[
M_{0q} \leq F(T_q, k_{0q}, \lambda_{0q}). 
\]

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 \( ith \) row and \( jth \) 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.
- \( M_{0i} \) 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 Aeronautical Propulsion 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
shown in Table III. The sensors with the largest fault signatures in descending order are $P_{30}$, $W_f$, $T_{30}$, $P_{15}$, $P_{24}$, $T_{24}$, $N_c$, and $T_{24}$. This gives eight potential sensor sets for detecting a fan fault: the first potential set uses only sensor $P_{30}$, the second potential set uses $P_{30}$ and $W_f$, 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 subject to the FPR constraint.

This process described in the previous paragraph was repeated for each fault shown in Table III. 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 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 experimental confusion matrices.
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 constraint.

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 approach has a condition number of 1.65, while the matrix in Table IX for the parity space approach has a condition number of 1.83. 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 \( y \) 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

\[
P(x > y) = \int_{-\infty}^{\infty} \left[ 1 - F_x(z) \right] f_y(z) dz.
\]

**Lemma 6**: If \( y = T + x \), where \( x \) is a random variable, and \( T \) is a constant, then

\[
f_x(w) = f_x(w - T)
\]

\[
F_y(w) = F_x(w - T).
\]

**Lemma 7**: If \( y = T - x \), where \( x \) is a random variable, and \( T \) is a constant, then

\[
f_y(w) = f_x(T - w)
\]

\[
F_y(w) = 1 - F_x(T - w).
\]

**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

\[
F_z(w) = \begin{cases} 
F_x(w) & w < T \\
1 & w \geq T 
\end{cases}
\]

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

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
\[ f_z(w) = \begin{cases} f_x(w) & w \geq T \\ 0 & w < T \end{cases} \quad f_z(w) = \begin{cases} f_x(w) & w \geq T \\ 0 & w < T \end{cases} \]
\[ f_z(w) = \begin{cases} f_x(w) & w \geq T \\ 0 & w < T \end{cases} \quad f_z(w) = \begin{cases} f_x(w) & w \geq T \\ 0 & w < T \end{cases} \]
\[ f_z(w) = \begin{cases} f_x(w) & w \geq T \\ 0 & w < T \end{cases} \]

Proof of Theorem 1: Equation (5) gives the definition of CNR as
\[ \text{CNR} = P[(S_1 < T_1), \ldots, (S_n < T_n)] \]
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
\[ \text{CNR} = P(S_1 < T_1) \cdots P(S_n < T_n) = \prod_{i=1}^{n} \text{TNR}(T_i, k_i). \]

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 $Y_m$ is a superset of $Y_i$ 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
\[ \text{CNR} = \text{TNR}(T_m, k_m). \]

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

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
\[ D_q = P(S_q - T_q > S_j - T_j \text{ for all } j \neq q) \geq \prod_{j \neq q} D_{q_j} \]
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.
\[ \text{CCR}_q = P[(S_q > T_q), (S_q - T_q > S_j - T_j) \text{ for all } j \neq q)] \]
\[ \leq \min_{j \neq q} P[(S_q > T_q), (S_q - T_q > S_j - T_j)] \]
\[ \leq \min_{j \neq q} \text{CCR}_{q_j}. \]

QED

Proof of Theorem 5: Given that we have $n > 2$ fault detection algorithms, the misclassification rate $M_{qj}$ is bounded from above by $M_{qj}^*$. So to obtain an upper bound for $M_{qj}$, we use one of (21)–(24) as appropriate. This approach gives
\[ M_{qj} = P[(S_j > T_j), (S_j - T_j > S_q - T_q) \text{ for all } i \neq j)] \]
\[ \leq P[(S_j > T_j), (S_j - T_j > S_q - T_q)] \]
\[ \leq M_{q_j}^*. \]

QED

Proof of Theorem 6: The probability $M_{qj}$ that no fault is detected when fault $q$ occurs is given as
\[ M_{qj} = P(S_j < T_j) \quad \text{for all } j \neq q \]
\[ \leq P(S_j < T_j) \leq P(T_q, k_q, \lambda_{q_j}). \]

QED

REFERENCES


