



# Sensor Data Qualification Technique Applied to Gas Turbine Engines

*Jeffrey T. Csank and Donald L. Simon  
Glenn Research Center, Cleveland, Ohio*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Information Desk at 443-757-5803
- Phone the NASA STI Information Desk at 443-757-5802
- Write to:  
STI Information Desk  
NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

NASA/TM—2013-216609



# Sensor Data Qualification Technique Applied to Gas Turbine Engines

*Jeffrey T. Csank and Donald L. Simon  
Glenn Research Center, Cleveland, Ohio*

National Aeronautics and  
Space Administration

Glenn Research Center  
Cleveland, Ohio 44135

---

December 2013

## Acknowledgments

The authors would like to thank William Maul, Christopher Fulton, and Thomas (Shane) Sowers for their guidance on implementing the Sensor Data Qualification and the Aviation Safety Program, Vehicle Systems Safety Technologies Project for funding this work.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

*Level of Review:* This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

National Technical Information Service  
5301 Shawnee Road  
Alexandria, VA 22312

Available electronically at <http://www.sti.nasa.gov>

# Sensor Data Qualification Technique Applied to Gas Turbine Engines

Jeffrey T. Csank and Donald L. Simon  
National Aeronautics and Space Administration  
Glenn Research Center  
Cleveland, Ohio 44135

## Abstract

This paper applies a previously developed sensor data qualification technique to a commercial aircraft engine simulation known as the Commercial Modular Aero-Propulsion System Simulation 40,000 (C-MAPSS40k). The sensor data qualification technique is designed to detect, isolate, and accommodate faulty sensor measurements. It features sensor networks, which group various sensors together and relies on an empirically derived analytical model to relate the sensor measurements. Relationships between all member sensors of the network are analyzed to detect and isolate any faulty sensor within the network.

## I. Introduction

Sensor fault detection and isolation algorithms increase the reliability of systems by enabling the early detection and removal of faulty data from the control system. Undetected sensor faults can impact the performance of closed loop systems that rely on these sensor measurements; therefore the ability to remove faulty sensor data prior to control action is desired to preserve the fidelity of the system.

The National Aeronautics and Space Administration (NASA) has previously explored sensor fault detection, isolation, and accommodation algorithms for jet engines. A survey of previous techniques has been provided by Merrill (Ref. 1). Fault detection and isolation schemes rely on redundant information to determine if a sensor measurement is valid. The redundant information can be a measurement obtained from a duplicate sensor, known as physical or direct redundancy, current or successive measurement samples from a single sensor, known as temporal redundancy, or estimated measurement information produced by a reference model which describe the expected behavior, known as analytical redundancy (Ref. 2). Common temporal redundancy checks include rate checks, which ensure that the derivative of the sensor data is less than a predetermined maximum value. Analytical redundancy relies on the use of models, such as linear mapping techniques and/or observers, to produce an estimated value that is used to determine if the data is acceptable.

The NASA advanced detection, isolation, and accommodation program focused on improving the overall system reliability of aircraft engines through the use of analytical redundancy methods (Refs. 3 and 4). The applied analytical redundancy method, referred to as an accommodation filter, uses a set of optimized engine estimates to isolate the faulty sensor. This approach can remove the faulty measurements from further use, and in some cases can replace the faulty measurements with estimated values as part of the accommodation algorithm. The estimated sensor measurements are created from a Kalman filter that incorporates a simplified engine model.

Another proposed approach is to simply select a “safe” measurement when two physically redundant channels disagree (Ref. 5). Consider a fan speed controlled engine that includes two physically redundant fan speed sensors, either of which may fail high or low. The existence of a sensor fault can be detected by comparing the measurements from the two sensors, but without an independent third channel it is not possible to isolate which of the two sensors is faulty. In such a scenario, the safe accommodation action would be to select the high sensor measurement for controlling the engine. If the fan speed sensor fails low, and the lower value is selected, the engine could over speed; resulting in a potentially catastrophic turbo machinery failure. However, if the larger value is selected and the fan speed sensor fails high, the engine will decrease its speed and power, but this is not a catastrophic event. It is noted in Reference 5,

that the effectiveness of this strategy deteriorates if the sensor fault is so large that the engine would shut down. To help avoid this situation, Reference 5 proposes a fuzzy-logic-based approach with the use of a third analytical measurement from a real time engine model.

Recently, NASA in conjunction with Expert Microsystem Inc., has developed a new sensor fault detection, isolation, and accommodation approach referred to as the Sensor Data Qualification (SDQ) system. SDQ has been designed for future automated and human in the loop space systems (Refs. 6 to 8). The SDQ approach features an analytical redundancy network to determine the state of the individual sensors in the network. Relationships are defined between all member sensors in the network. The number of failed relationships for each member sensor is tracked and used to detect and isolate the faulty sensor; the number of failed relationships must be greater than some number, which is predetermined and based on the number of validated sensors. Once a sensor is identified as faulty, the sensor is removed from the network.

The work presented in this paper focuses on applying the analytical network of SDQ to a commercial aircraft engine simulation with the goal of diagnosing and accommodating faulty sensor measurements to preserve the accuracy of the sensor measurement reported to the controller. Due to weight restrictions, only dual channel systems are being considered in this study. This study assumes that the sensors are fully operational at the beginning of a flight and is limited to a single sensor failure. Section II discusses the SDQ approach. The commercial aircraft simulation that is being used to demonstrate and test the SDQ approach is discussed in Section III. Tuning of the SDQ algorithms is discussed in Section IV. Simulation results are shown in Section V with conclusions in Section VI.

## Nomenclature

$A$	Reference made to the first channel of a redundant system
ARC	Analytical Redundant Channel
$B$	Reference made to the second channel of a redundant system
C-MAPSS40k	Commercial Modular Aero-Propulsion System Simulation 40,000
$i$	Measurement grouping, such as Fan speed sensors
$t_{ijarc}$	Analytical redundancy check limit between measurement $i$ and measurement $j$
$t_{imax}$	Maximum limit for measurement $i$
$t_{imin}$	Minimum limit for measurement $i$
$t_{irate}$	Maximum limit for the derivative of measurement $i$
$t_{ircc}$	Redundant channel check limit for measurement $i$
$j$	Measurement $j$ of an analytical redundancy network
$k$	Analytical model linear relationship factor
NASA	National Aeronautics and Space Administration
$Nc$	Core speed (rpm)
$NcR$	Corrected core speed (rpm)
$Nf$	Fan speed (rpm)
$NfR$	Corrected fan speed (rpm)
PLA	Power Lever Angle (degrees)
$Ps3$	High pressure compressor discharge static pressure (psi)
$P2$	Inlet pressure (psi)
$P50$	Low pressure turbine discharge pressure (psi)
$rpm$	Revolutions per minute

$r_{ixjy}$	Residual for analytical network relation between measurements $i$ sensor channel $x$ and $j$ sensor channel $y$
SDQ	Sensor Data Qualification
$T2$	Inlet temperature (° Rankine)
$T30$	High pressure compressor discharge temperature (° Rankine)
$T50$	Low pressure turbine discharge temperature (° Rankine)
$x$	Reference to a channel of sensor $i$
$y$	Reference to a channel of sensor $j$

## II. Application of Sensor Data Qualification

The Sensor Data Qualification (SDQ) application is a multi-layered system consisting of reasonableness checks, analytical redundancy checks, and additional logic to minimize false alarms. One factor in the reduction of false alarms, for this application, is accomplished by the execution of individual sensor checks before performing more complex analytical redundancy checks. The analytical redundancy checks utilize multiple sensors as inputs and are only processed once all the incoming data has been determined to be reasonable by the individual sensor checks. This sequence of operations minimizes the impact of corrupted data on the analytical networks. A high level flowchart showing the information flow through the components that comprise the SDQ architecture is shown in Figure 1.

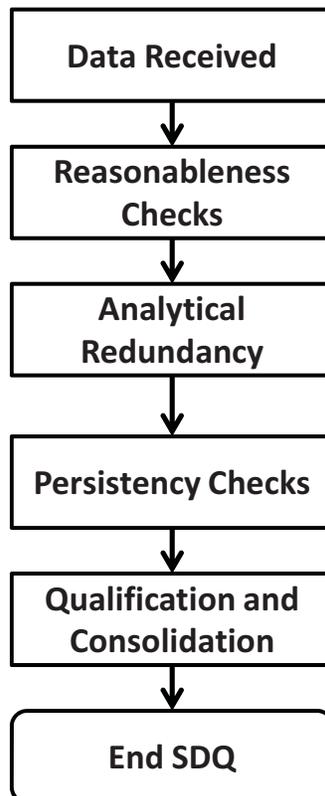


Figure 1.—Flow chart showing the high level operation of the applied SDQ algorithm.

During each time step, or execution, the SDQ algorithm receives the incoming sensor measurement data and then performs the reasonableness checks, (i.e., range and rate checks), on each individual sensor. If a reasonableness check fails for an individual sensor, the sensor is flagged for the current time step. If a sensor is flagged, no other calculations involving the sensor will be performed during the current time step. After the reasonableness checks, the analytical redundancy checks are performed for networks that contain sufficient qualified and non-flagged sensors to assess the remaining network sensors. The persistency check requires that a sensor be flagged for a defined number of consecutive executions prior to being disqualified. This is done to prevent a sensor being disqualified for random bad data points, or outliers. In this application, each individual sensor has an associated strike counter to track the number of consecutive executions that it has been flagged. An assumption is made that only a single sensor can be flagged as faulty at any given time step, otherwise detection of multiple sensor faults by the SDQ checks is an indication of a process change within the system. If a single sensor is flagged, the corresponding strike counter is incremented during the persistency checks. If the strike counter reaches its limit, the sensor is disqualified. If, prior to reaching its strike counter limit, a qualified sensor is not flagged on any subsequent time step its strike counter is reset to zero. In addition, if multiple sensors are flagged as faulty during a single time step, the strike counter for all qualified and flagged sensors are reset to zero as well. In this situation, it is assumed that the engine is operating at a mode outside the region the SDQ network of checks was designed to monitor and the data is good. The final step of SDQ is the qualification and consolidation subsystem, which is responsible for qualifying and consolidating the sensor data. During normal operation with no flagged or disqualified sensors, the data reported to the controller in this study is the average value of the Channels *A* and *B* sensor measurements. Once a sensor is identified as disqualified by reaching the strike count limit, the sensor measurement is removed from all further calculations. In this case, the value reported to the controller would be the measurement from the remaining good channel. Note that in this study, the possibility of more than one sensor fault is not considered. The remainder of this section will discuss the algorithms used for the reasonableness checks and the analytical redundancy checks.

## A. Reasonableness Checks

The reasonableness checks are designed to identify gross faults such as a short or open circuit in the sensor path that result in an unrealistic sensor measurement. Depending on the sensor, these faults can manifest themselves as a very large negative or positive measurement, a zero, or any other measurement that would not be physically possible, for example a negative pressure or temperature measurement in the turbine section of an engine. Reasonableness checks establish limits or thresholds to be applied to each data sample from the sensor. These limits can be established at the extreme measurement ranges of the sensor or they can be adjusted to detect a measurement outside the operational envelope resulting in quicker detection, but also increasing the potential for a false detection. Two common checks to protect against these types of faults are range checks and rate checks.

The range check ensures that the incoming sensor data each time step is within known minimum and maximum limits, or:

$$t_{i \min} \leq i_{A/B} \leq t_{i \max} \quad (1)$$

Where *i* is the measurement, such as the compressor discharge pressure, compressor discharge temperature, etc., the subscripts *A/B* refers to the sensor channels being tested, both channels are tested separately denoted by (*A/B*), and  $t_{i \min}$  and  $t_{i \max}$  are the minimum and maximum limits for the measurement, *i*. The range check is mainly intended to detect a hard sensor fault, such as a short or open circuit, when the value being reported is outside the expected range.

The rate check ensures that the absolute value of the incoming data's derivative is below some maximum limit:

$$\left| \frac{di}{dt} i_{A/B} \right| \leq t_{irate} \quad (2)$$

Where  $t_{irate}$  is the maximum derivative possible for the measurement  $i$  and can be implemented as either a time base derivative or change from the previous sample. This limit could be based on the physical capability of the physical sensor, or determined empirically from data. The rate check can aid in detecting small magnitude step changes, extremely noisy sensor signals, and intermittent signal behavior that can be due to either a loose connector or failing sensor.

## B. Analytical Redundancy Checks

The analytical redundancy checks are intended to identify small magnitude and slow degrading sensor faults. These types of faults fall within the detection thresholds of the previously described reasonableness checks. The analytical redundancy check applied in this study is adapted from the SDQ technology that has been implemented for several space applications (Refs. 6 to 8). The SDQ analytical redundancy checks contain networks of sensors that have a strong relationship with one another. When all sensors within a given network are operating nominally, each of the sensor relationships are expected to be satisfied. However, in the event of a sensor fault, the faulty sensor can be identified by assessing which relationships are satisfied and which relationships are violated. A network relationship could involve two or more sensors, but in practice relationships typically involve only two sensors and while non-linear relationships are possible, linear relationships are sufficient, easier to develop and verify. These relationships are established either analytically or empirically based on derived knowledge of the system or process being controlled and available data. One method of defining the grouping of sensors and sensor relationships that comprise a sensor network is to use an automated software product known as SureSense. The SureSense Data Quality Validation Studio, developed by Expert Microsystems in conjunction with NASA Glenn Research Center, derives empirical relations for the network based on data through statistical analysis, pattern recognition, and neural networks. In this study, parameter relationships derived from an analytical model will be used to construct the sensor networks as described below.

The first step in the analytical redundancy check is to determine if there is a mismatch between the two physically redundant sensors (Channels  $A$  and  $B$ ):

$$\left| i_A - i_B \right| < t_{ircc} \quad (3)$$

where  $t_{ircc}$  is the redundant channel check limit for measurement  $i$ . If the relationship of Equation (3) is valid, then no further calculations are required since the redundant sensors are in agreement. However, if the relationship of Equation (3) fails, then additional checks are performed to determine which of the two physically redundant sensors is faulty. This is done by assessing the defined relationships that exist between measurement  $i$  and the other measurements included in the same network. These other sensed measurements are referred to here as relation sensors. The difference, or residuals, between each sensor  $i$  and its defined relationships are calculated for all member sensors of the network. For an analytical network comprised of two measurements,  $i$  and  $j$ , each with a redundant sensor Channels  $A$  and  $B$ , the residuals become:

$$\begin{aligned} r_{iAjA} &= i_A - k_{ij} j_A \\ r_{iBjA} &= i_B - k_{ij} j_A \\ r_{iAjB} &= i_A - k_{ij} j_B \\ r_{iBjB} &= i_B - k_{ij} j_B \end{aligned} \quad (4)$$

where  $k$  is the linear relationship between measurement  $i$  and measurement  $j$ , which can be found from a linear model, piece-wise linear model, etc. Note that for each measurement  $i$ , there could be more than one relation measurement  $j$ , although only one is shown for the example presented in Equation (4). Each calculated residual is then compared to an analytical redundancy check limit ( $t_{ijarc}$ ) to assess whether the defined relationship between measurements  $i$  and  $j$  is satisfied:

$$|r_{ixjy}| < t_{ijarc} \quad (5)$$

Where  $x$  is the measurement  $i$  sensor channel and  $y$  is the relation measurement  $j$  sensor channel. For the example presented in Equation (4), a single measurement  $i$  sensor channel is flagged only if, that sensor fails its redundant channel check, both analytical redundancy checks containing the sensor fail, and at least one of the residuals of the measurement  $i$  redundant sensor channel satisfies its analytical redundancy check, or:

$$\begin{aligned} \text{fail } i_A \text{ if } |i_A - i_B| \geq t_{ircc} \text{ and } |r_{iAjA}| \geq t_{ijarc} \text{ and } |r_{iAjB}| \geq t_{ijarc} \text{ and } \left( |r_{iBjA}| < t_{ijarc} \text{ or } |r_{iBjB}| < t_{ijarc} \right) \\ \text{fail } i_B \text{ if } |i_A - i_B| \geq t_{ircc} \text{ and } |r_{iBjA}| \geq t_{ijarc} \text{ and } |r_{iBjB}| \geq t_{ijarc} \text{ and } \left( |r_{iAjA}| < t_{ijarc} \text{ or } |r_{iAjB}| < t_{ijarc} \right) \end{aligned} \quad (6)$$

Additional logic is added to ensure that not more than one sensor can be identified, or flagged, as faulty from a single network at a current time step. In the case where more than one sensor is flagged as faulty, no sensor would be identified. This is to guard against incorrectly disqualifying a good sensor in the event that the system is operating outside its normal operational envelope.

### III. Commercial Aircraft Engine Simulation Testbed

In this study, the SDQ fault detection, isolation, and accommodation system is applied to a commercial aircraft engine simulation known as the Commercial Modular Aero-Propulsion System Simulation 40,000 (C-MAPSS40k). C-MAPSS40k is a 40,000 lb thrust class, two spool, physics-based, component level, high bypass turbofan engine simulation and closed loop controller modeled in the Matlab/Simulink (The MathWorks, Inc.) environment (Refs. 9 and 10). The application of SDQ to the C-MAPSS40k architecture is shown in Figure 2. To simulate and test the SDQ approach, the C-MAPSS40k simulation is modified to allow for dual channel operation. A second set of sensor measurements, referred to as Channel B, has been added to the original set of sensors, Channel A, which

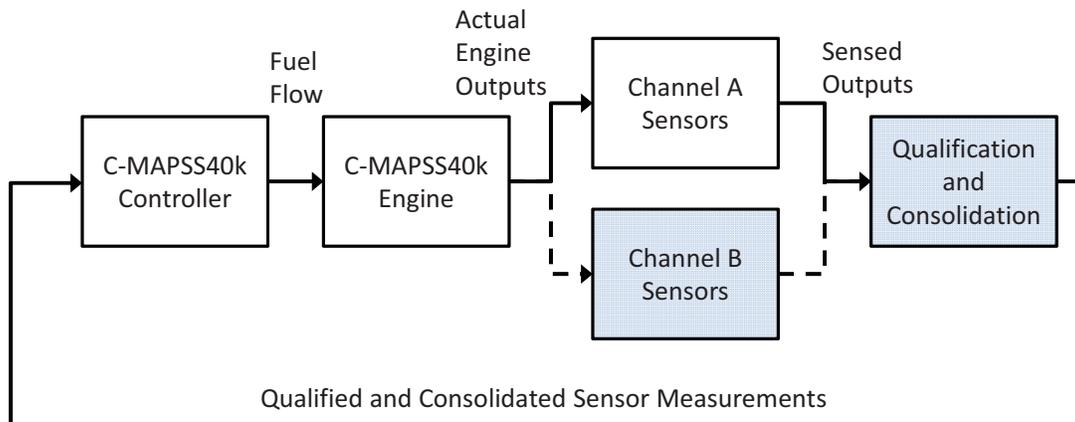


Figure 2.—A block diagram of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS40k) configured for dual channel operation. The addition of a second set of sensor measurements and qualification and consolidation subsystems are shaded gray and the additional signal path is shown as a dashed line.

include the following measurements: high pressure compressor discharge static pressure and total temperature ( $Ps3$  and  $T30$ ), low pressure turbine discharge total pressure and total temperature ( $P50$  and  $T50$ ), fan speed ( $Nf$ ) and core speed ( $Nc$ ). The Channels  $A$  and  $B$  measurements are inputs to the qualification and consolidation subsystem. The qualification and consolidation subsystem calculates the average of the qualified Channels  $A$  and  $B$  sensors, and provides a single consolidated vector of measurement values as feedback to the controller. The goal of SDQ is to correctly detect and isolate a single sensor fault on any of the engine's dual channel sensors within the same flight that the fault occurred.

## IV. Algorithm Tuning

To determine the limits for SDQ, a database of 500 flights at random takeoff conditions and 500 flights at random cruise conditions with no faults are created. The range of the takeoff and cruise flight conditions are shown in Table 1. Each flight is a 30 sec simulation with two throttle movements. Half of the simulations at each flight condition consist of a burst and chop, where the simulation starts at the throttle low position, moves to the throttle high position at a simulation time of 10 sec, and returns to the throttle low position at a simulation time of 20 sec. The other simulations are a chop and burst, where the simulation starts in the throttle high position, moves to the throttle low position, and returns to the throttle high position with the same timing. White noise is added to the analog pressure and temperature sensor signals.

### A. Reasonableness Checks

The range and rate check limits for each measurement is established by analyzing the nominal data to determine the extreme values, and setting the limits accordingly. The minimum range limit for a given measurement is determined by scaling the minimum value for all 1,000 flights by 0.75. Likewise the maximum range limit is determined by scaling the maximum value for all 1,000 flights by 1.25. The additional 25 percent scaling ensures that the range checks will not disqualify or remove data unless it is absolutely certain the data is out of range. The rate limit for each measurement is established by finding the maximum absolute value of the sample change of each measurement for all 1,000 flights. The range (min and max) and rate limit values are shown in Table 2 for the six sensor measurement types considered in this study.

TABLE 1.—THE POSSIBLE RANGE FOR THE RANDOM TAKEOFF AND CRUISE NOMINAL FLIGHTS

	Takeoff	Cruise
Altitude, ft	0 to 3,500	27,000 to 36,000
Mach number	0.16 to 0.25	0.55 to 0.78
Delta ambient temperature, °R	-20 to 40	-5 to 5
Engine degradation	50 hr to end of life	50 hr to end of life
Throttle low, degrees	41 to 44 (Idle)	54 to 58 (Flight idle)
Throttle high, degrees	76 to 80 (Takeoff)	60 to 68 (Cruise)

TABLE 2.—THE REASONABLENESS CHECKS LIMITS DETERMINED FROM THE NOMINAL FLIGHT DATA AFTER SCALING

Sensor measurement type	Min limit, ( $t_{min}$ )	Max limit, ( $t_{max}$ )	Rate, ( $t_{rate}$ )
$Nf$	1233, rpm	5325, rpm	20.8, rpm/sample
$Nc$	6763, rpm	15594, rpm	42.4, rpm/sample
$Ps3$	52, psi	579, psi	5.29, psi/sample
$P50$	3.4, psi	31.5, psi	0.1292, psi/sample
$T30$	726.7, °R	1966, °R	4.4, °R/sample
$T50$	678, °R	2170, °R	14.2, °R/sample
$Nf$	1233, rpm	5325, rpm	20.8, rpm/sample

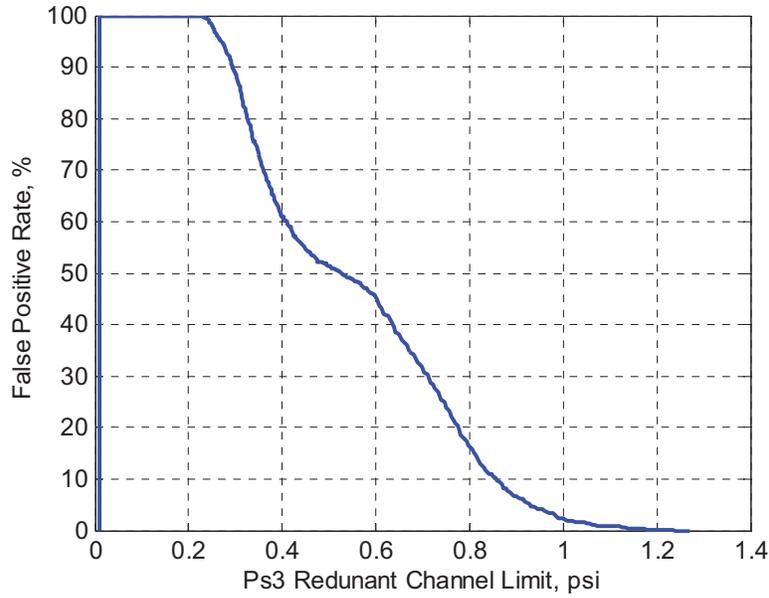


Figure 3.—The relationship between the  $Ps3$  redundant channel check limit,  $Ps3_{rcc}$ , and the false positive rate.

## B. Analytical Redundancy Checks

The analytical redundancy checks actually consist of a redundant channel check and an analytical network check based upon a sensor network (i.e., a defined grouping of sensors along with their defined interrelationships). Both of these checks are tuned independently, but collectively affect the performance of the analytical redundancy check. The redundant channel check, Equation (3), is the difference between the Channels  $A$  and  $B$  measurement. The redundant channel check limit,  $i_{rcc}$ , is set to maintain a maximum false positive rate (0.005 percent), the number of flights that the check is incorrectly violated divided by the total number of flights in which a fault was not present, and is determined from analyzing the nominal flight data. Simulations are executed using all the nominal flight data to determine the redundant channel check limit to meet the false positive rate requirement with the persistency check requirements (set to 3 consecutive strikes), reducing the number of false alarms. Figure 3 shows the relationship between the  $Ps3$  redundant channel check limit and the false positive rate. As the  $Ps3$  sensor redundancy check limit decreases, smaller magnitude differences will be detected or flagged, but the false positive rate will also increase.

The analytical network checks are more complex and require development of relationships between measurements. The measurements to be included in the analytical networks are:  $Nf$ ,  $Nc$ ,  $Ps3$ ,  $P50$ ,  $T30$ , and  $T50$ . In this implementation, it was chosen to have three fixed networks that consist of two different sensor measurements which are denoted as  $i$  and  $j$ . Three networks were designed, one for rotor speeds ( $i = Nf$  and  $j = Nc$ ), one for pressure sensors ( $i = Ps3$  and  $j = P50$ ), and one for temperature sensors ( $i = T30$  and  $j = T50$ ). Grouping the sensors into networks based on the measurement type reduced the limits in the defined relationships between sensors due to differences in sensor dynamics and thereby provided increased sensitivity to sensor fault detection. Using a single  $i$  and  $j$  relationship of the analytical redundancy network, Equation (6), for the qualification of either measurement simplified the implementation. The reasonableness checks are executed for both  $i$  and  $j$  sensors and if any sensor fails these checks, then the associated analytical redundancy network processing is suspended. For the analytical redundancy checks, in this form, cross checks are performed on both  $i$  and  $j$  sensors and if one and only one of the measurements fails its' cross check, then the remaining analytical redundancy network checks are applied to resolve the failed sensor. For example of the rotor speed network, if the  $Nc$  cross checks fail, the same analytical network relationships are used to determine the failed  $Nc$  sensor as would have been applied to

the  $Nf$  sensor failing its cross check. Equation (6) can be modified to return the failed  $j$  sensor if the  $j$  sensor fails the redundant channel check:

$$\begin{aligned} \text{fail } j_A \text{ if } |j_A - j_B| \geq t_{jrcc} \text{ and } |r_{iAjA}| \geq t_{ijarc} \text{ and } |r_{iBjA}| \geq t_{ijarc} \text{ and } \left( |r_{iAjB}| < t_{ijarc} \text{ or } |r_{iBjB}| < t_{ijarc} \right) \\ \text{fail } j_B \text{ if } |j_A - j_B| \geq t_{jrcc} \text{ and } |r_{iAjB}| \geq t_{ijarc} \text{ and } |r_{iBjB}| \geq t_{ijarc} \text{ and } \left( |r_{iAjA}| < t_{ijarc} \text{ or } |r_{iBjA}| < t_{ijarc} \right) \end{aligned} \quad (7)$$

Other options were considered for designing the analytical networks. One option was to group the sensors into two networks based on whether the sensor provided a measurement from either a low or high rotor location. For example, sensors for the low pressure compressor and low pressure turbine would be grouped with the fan speed sensor measurement, but the data showed that there would be larger limits in the defined sensor relationships based on this type of network, mainly due to a difference in dynamics associated with the sensors.

Using the three network approach described above, there are two measurements in each network and therefore would resemble the network defined in the Equation (4). For example, the rotor speed network consists of both fan speed and core speed measurements. The goal is to determine a linear relationship factor ( $k_{ij}$ ) that minimizes the difference between the two measurements, across all four sensor combinations. Figure 4 shows the relationship between corrected fan speed ( $NfR$ ) and corrected core speed ( $NcR$ ), shown as  $NfR/NcR$  on the y-axis and corrected fan speed on the x-axis for 20 of the 1,000 nominal flights, displayed as the green region. Figure 4 also shows an approximation of the nominal flight data, determined using a least squares fit, that is to be used as the linear relationship factor,  $k_{ij}$ . The determined linear relationship factors,  $k_{ij}$ , are then archived to facilitate retrieval during SDQ execution via first order linear interpolation scheduled based on the corrected fan speed. For this application, the corrected fan speed value used for scheduling the interpolation is calculated using the average fan speed and average inlet temperature:

$$NfR = \frac{0.5(NfA + NfB)}{\sqrt{0.5(T2A + T2B)}/518.67} \quad (8)$$

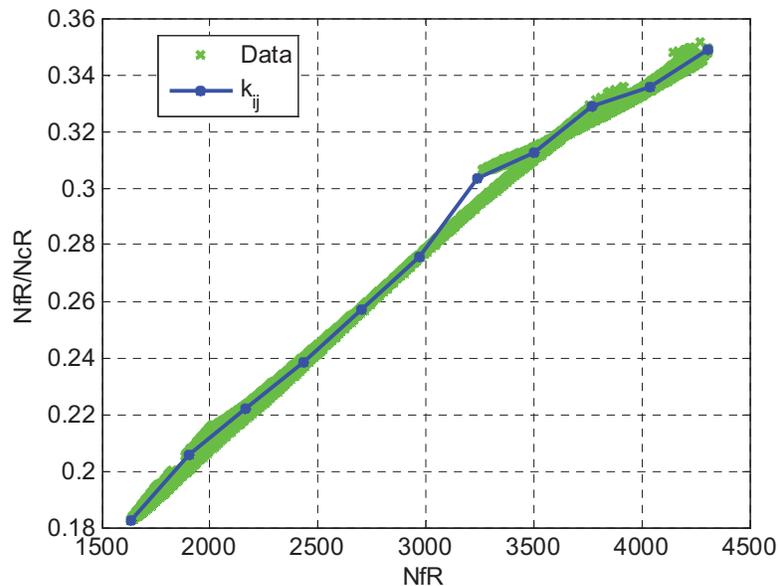


Figure 4.—Example relationship between the linear relationship factor ( $k_{ij}$ ) and nominal data based 20 flights for the rotor speed network.

The average corrected fan speed calculation shown in Equation (8) is applied to retrieve the  $k_{ij}$  information for all sensor types except for the fan speed sensors themselves. Note that the fan speed sensor affects all three networks since the model is based on corrected fan speed. For an individual fan speed sensor channel, only the opposite fan speed channel is used in scheduling the interpolation (not the average of both channels). This is done to guard against a fault in a fan speed sensor channel corrupting the calculation of the interpolation scheduling parameter. For example, a slow sensor drift in fan speed Channel  $A$  may be difficult to detect since the linear relationship factor will tend to follow the fault and may result in the good sensor being disqualified. Therefore, when calculating the Channel  $A$  fan speed residual, the linear relationship factor will be retrieved using the Channel  $B$  fan speed measurement only.

During testing, it was observed that the incorrect fan speed sensor channel was disqualified 25 to 33 percent of the time for a negative slow fan speed sensor drift. Additional logic was added to help address this issue. The additional logic is designed to flag a Channel  $A$   $Nf$  fault when the two Channel  $A$  residuals are less than the analytical model limit and the two Channel  $B$  residuals are greater than the analytical model limit, and the residual between  $Nf$  Channels  $A$  and  $B$  is less than the redundant channel check limit. The opposite of this is flagged as a Channel  $B$  fault, or:

$$\begin{aligned} \text{fail } i_A & \text{ if } i_A - i_B < -t_{ircc} \text{ and } r_{iAjA} < -t_{ijarc} \text{ and } r_{iAjB} < -t_{ijarc} \text{ and } r_{iBjA} > t_{ijarc} \text{ and } r_{iBjB} > t_{ijarc} \\ \text{fail } i_B & \text{ if } i_A - i_B > t_{ircc} \text{ and } r_{iAjA} > t_{ijarc} \text{ and } r_{iAjB} > t_{ijarc} \text{ and } r_{iBjA} < -t_{ijarc} \text{ and } r_{iBjB} < -t_{ijarc} \end{aligned} \quad (9)$$

This change is necessary since the linear relationship factor,  $k_{ij}$ , is determined using the corrected fan speed. A  $Nf$  fault affects the model relationships and will cause all the residuals in the network to fail.

With the sensor network and relationships defined, the analytical redundancy check limit ( $i_{jarc}$ ) can be found. Ideally, the analytical redundancy check limit would be set to achieve a maximum false positive rate based on the nominal data; similar to the redundant channel check. The relationship between the analytical redundancy check limit and the false positive rate for the  $Ps3$  sensor is shown in Figure 5. In Figure 5, the false positive rate does not always increase as the limit value decreases, and the false positive rate does not approach 100 percent as the limit reaches zero. Using the network approach requires that one sensor is identified as faulty against three other sensors, and if more than one sensor is

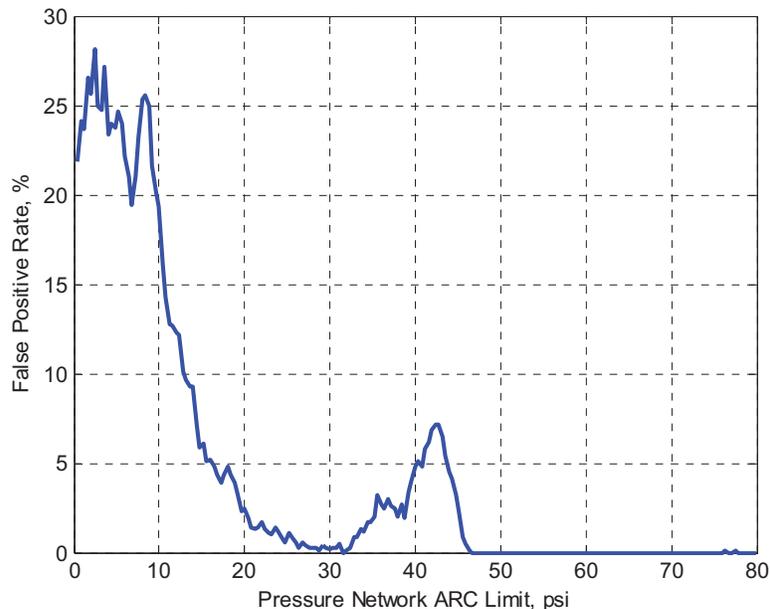


Figure 5.—The relationship between the false positive rate and the analytical limit for the pressure sensor network limit using the  $Ps3$  sensor.

TABLE 3.—THE ANALYTICAL NETWORK LIMITS

Network	Member sensor	Redundant channel check limit	Analytical redundancy check limit
Rotor speed	$i = Nf$ , rpm	1.0	162.0
	$j = Nc$ , rpm	1.0	
Pressure	$i = Ps3$ , psi	1.2341	10.0
	$j = P50$ , psi	0.0513	
Temperature	$i = T30$ , °R	1.145	266.4
	$j = T50$ , °R	1.168	

identified as faulty, then no sensor is identified and the strike counters for all sensors are reset. As the analytical redundancy check limit decreases, there is more chance that the strike counters are reset due to more than one sensor failing the relationship; which explains the decrease in the false positive rate as the limit value decreases from 42 to 30.

The approach taken in this study is to establish the analytical redundancy check limit to (1) minimize the false positive rate and (2) maximize the detection rate of a single sensor drift fault; for example a 15 percent deviation of a single channel, which is slow enough not to be detected by the rate check. This approach allows for the limits to be set to include all known good nominal data and maximize the detection rate of faults of a target value. To start with, limits were adjusted to ensure that there were zero false alarms for the nominal 1,000 flight conditions and 100 percent corrected detections of a 15 percent deviation of the sensors from the pre-fault sensor value. Closed-loop simulations were run at a subset of the 1,000 nominal flight conditions, in this case 12 was arbitrarily chosen, to test the analytical redundancy limits and adjusted to maximize the correct detection rate. This process is repeated for all sensors in each network. The analytical network limits, which include the redundant channel check limits (non-adjusted) and analytical redundancy check limits (adjusted), are shown in Table 3.

## V. Simulation and Results

With the SDQ system designed, three different types of faults are inserted into the closed loop C-MAPSS40k simulation: a step change, a slow sensor drift, and an intermittent signal. The sensor fault input profiles, shown as a percentage of the sensor fault magnitude, applied to generate the sensor faults are shown in Figure 6. The step and drift faults can be in the positive direction or in the negative direction. The step fault occurs within 0.015 sec, 1 time step, whereas the sensor drift takes 50 sec to reach its full fault value. The intermittent fault is only in the negative direction due to the expected characteristic of this type of sensor failure and the applied time period varies over 2 to 15 time steps. Sensor faults are added to the system in the respective Channel *A* or Channel *B* subsystem shown in Figure 2. The faults will be inserted for both a takeoff flight condition and a cruise flight condition, shown in Table 4. The fault magnitudes tested are 1, 5, 10, 15, 20, 25, and 30, for each sensor and each channel (6 sensor types and 2 channels).

To test and demonstrate the benefit of the analytical redundancy checks, a baseline approach consisting of the reasonableness checks is compared to the full SDQ algorithm containing both the reasonableness and analytical redundancy checks. These algorithms were implemented and evaluated in the qualification and consolidation subsystem shown in Figure 2, and the results were documented in a confusion matrix using the Table 4 flight conditions. The rows of the confusion matrix represent the actual faulty sensor and the columns represent the detected and isolated fault. Ideally, the confusion matrix would represent an identity matrix where the diagonal would be equal to 1.0, 100 percent detection and isolation, corresponding to perfect detection and isolation of the fault. Note that SDQ was tuned for a 15 percent magnitude fault; therefore if SDQ only detects and isolates sensor faults  $\geq 15$  percent faults, then the correct diagnosis rate should be 57 percent (4 of the 7 fault magnitudes considered).

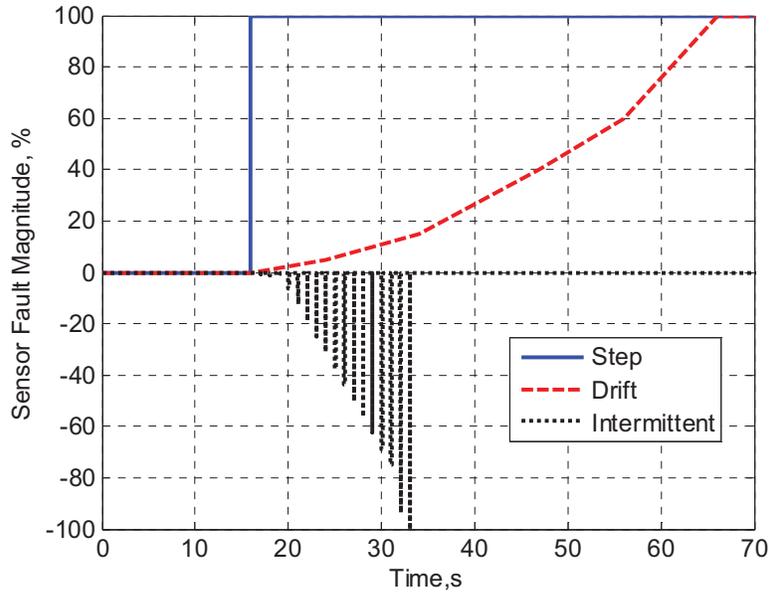


Figure 6.—Sensor fault input profiles. The input profiles show the percentage of the actual fault magnitude as a function of time. The drift failure was randomly chosen such that its derivative is below the sensor rate failure check limit.

TABLE 4.—THE SENSOR DATA QUALIFICATION TESTING FLIGHT CONDITIONS

	Takeoff	Cruise
Altitude, ft	791	33200
Mach number	0.165	0.725
Delta ambient temperature, °R	0	0
Engine degradation	End of life	50 hr engine
Throttle low, degrees	42	56
Throttle high, degrees	80	68

The confusion matrix for a step fault added to the simulation for the baseline data qualification approach is shown in Table 5. Note that in there is an additional column labeled “No,” which is the case in which the fault is undetected and allows the rows to sum to 1.0. All the step faults can be correctly detected >89 percent of the time. Note that when the fault is inserted into the Channel *B* sensors, the Channel *B* Detected and Isolated Faults are exactly the same as the Channel *A* detected and isolated faults for a Channel *A* actual fault. In addition, the Channel *B* detected and isolated faults for a Channel *A* Actual Fault are exactly the same as the Channel *A* detected and isolated faults for a Channel *B* actual fault. Therefore, only the results for the actual Channel *A* faults will be shown for the rest of the paper, since it clearly captures the performance. The confusion matrix for a Channel *A* step fault added to the SDQ simulation is shown in Table 6. All the step faults can be correctly detected >89 percent of the time, however the SDQ algorithm can correctly detect *Ps3* and *T50* faults better than the baseline. During testing, it was observed that the step faults are detected by the range checks and analytical redundancy checks. The rate checks were able to flag the step change in the incoming data, however since the transition occurred in one time step, the rate check would only flag the measurement for one time step and is not enough to meet the persistency requirement.

Another method to compare the baseline and SDQ data qualification techniques is to analyze the size of the fault when the fault is detected (detection magnitude) and the time from when the failure occurs to when the failure is detected (time to detection). A comparison of these two metrics for the *N<sub>f</sub>* step sensor failure, for both channels, all the tested flight conditions, and fault magnitude sizes, is shown in Figure 7. Figure 7 shows that the sensor step fault detection occurs in one time step (0.015 sec) for both approaches, further indicating that the step change (common to both approaches) is responsible for detecting a step sensor failure.

TABLE 5.—CONFUSION MATRIX FOR A STEP SENSOR FAULT WITH THE BASELINE APPROACH

		Detected and Isolated Faults												No	
		Channel A						Channel B							
		<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>		
Actual Fault	Channel A	<i>Nf</i>	0.982	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.018
		<i>Nc</i>	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		<i>Ps3</i>	0.000	0.000	0.857	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.143
		<i>P50</i>	0.000	0.000	0.000	0.929	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.071
		<i>T30</i>	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.893	0.000	0.000	0.000	0.000	0.000	0.000	0.107
Actual Fault	Channel B	<i>Nf</i>	0.000	0.000	0.000	0.000	0.000	0.982	0.000	0.000	0.000	0.000	0.000	0.018	
		<i>Nc</i>	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	
		<i>Ps3</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.857	0.000	0.000	0.000	0.143	
		<i>P50</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.929	0.000	0.000	0.071	
		<i>T30</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.893	0.107	

TABLE 6.—CONFUSION MATRIX FOR A STEP SENSOR FAULT IN CHANNEL A WITH SDQ

		Detected and Isolated Faults												No
		Channel A						Channel B						
		<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	
Actual Fault	Channel A	<i>Nf</i>	0.982	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.018
		<i>Nc</i>	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		<i>Ps3</i>	0.000	0.000	0.893	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.107
		<i>P50</i>	0.000	0.000	0.000	0.929	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.071
		<i>T30</i>	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.911	0.000	0.000	0.000	0.000	0.000	0.036

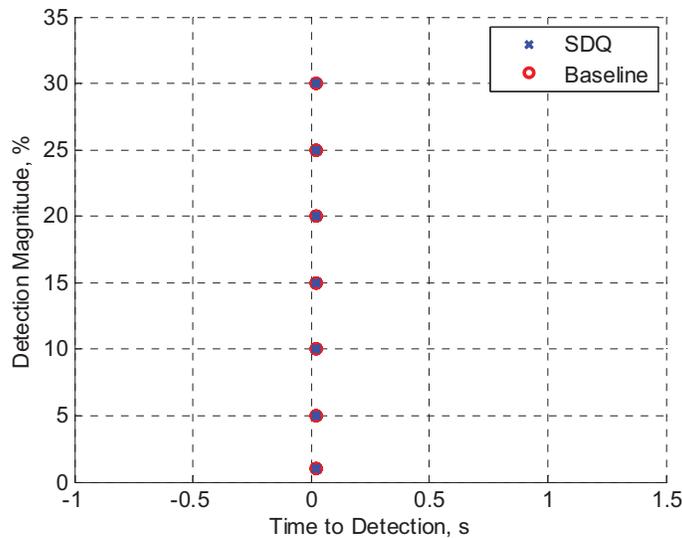


Figure 7.—Plot comparing the *Nf* step sensor fault detection magnitude (size of the fault when detected) and the time to detection for both the baseline and SDQ approaches.

The confusion matrix for a sensor drift fault for both the baseline approach and SDQ is shown in Table 7. The baseline approach could only detect 18 percent of the *Nf*, *Nc*, and *T30* slow sensor drift failures. The Table 7 results indicate that SDQ provides detection rates of >75 percent for the rotor speed faults, >51 percent for the pressure sensors and >33 percent for the temperature sensors. In regards to SDQ, the slow sensor drift fault, applied in this study, is detected by either the range check when the drift causes the sensor value to exceed an operational limit, or the analytical redundancy check.

Figure 8 compares the baseline and SDQ data qualification techniques for the detection magnitude and time to detection for the *Nf* sensor drift failure, for all the tested flight conditions and fault magnitude sizes. Figure 8 shows the SDQ technique is able to identify sensor drift failures at magnitudes between 2 to 10 percent at times between 10 to 52 sec, but is unable to detect the 1 percent sensor drift failures. The data shown in Figure 8 along with the confusion matrix shown in Table 7, demonstrates the benefit of adding the analytical redundancy network over just the reasonableness checks.

TABLE 7.—CONFUSION MATRIX FOR A SLOW SENSOR DRIFT FAULT FOR CHANNEL A COMPARING THE BASELINE APPROACH AND SDQ

Actual Fault		Detected and Isolated Faults												
		Channel A						Channel B						No
		<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	
Baseline	<i>Nf</i>	0.018	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.982
	<i>Nc</i>	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.982
	<i>Ps3</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
	<i>P50</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
	<i>T30</i>	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.982
	<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
SDQ	<i>Nf</i>	0.768	0.000	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.214
	<i>Nc</i>	0.000	0.750	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250
	<i>Ps3</i>	0.000	0.000	0.607	0.000	0.000	0.000	0.000	0.000	0.107	0.000	0.000	0.000	0.286
	<i>P50</i>	0.000	0.000	0.000	0.518	0.000	0.000	0.000	0.000	0.000	0.089	0.000	0.000	0.393
	<i>T30</i>	0.000	0.000	0.000	0.000	0.357	0.000	0.000	0.000	0.000	0.000	0.179	0.000	0.464
	<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.339	0.000	0.000	0.000	0.000	0.000	0.179	0.482

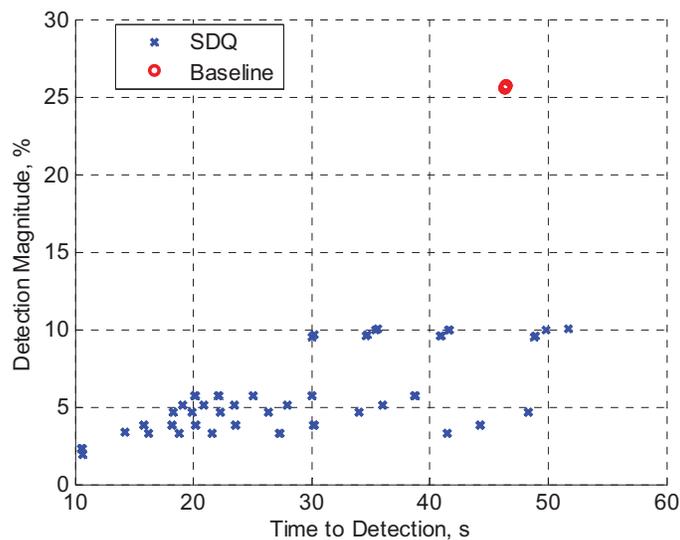


Figure 8.—Plot comparing the *Nf* sensor drift fault detection magnitude (size of the fault when detected) and the time to detection for both the baseline and SDQ approaches.

The confusion matrix for an intermittent fault is shown in Table 8, where both the baseline and SDQ approach can correctly identify a fault >78 percent. During the development and testing, it was observed that detecting the intermittent fault, especially the one designed for this work, is a function of the rate check. The intermittent fault applied in this work changes quickly, but can range from 2 to 15 time steps. Figure 9 compares the detection magnitude and time to detection for the  $N_f$  intermittent sensor failures, which shows that both the baseline and SDQ techniques perform the same, which should be expected since both techniques use the same rate check algorithm.

TABLE 8.—CONFUSION MATRIX FOR A INTERMITTENT SENSOR FAULT FOR CHANNEL A COMPARING THE BASELINE APPROACH AND SDQ

		Detected and Isolated Faults												No	
		Channel A						Channel B							
		$N_f$	$N_c$	$P_{s3}$	$P_{50}$	$T_{30}$	$T_{50}$	$N_f$	$N_c$	$P_{s3}$	$P_{50}$	$T_{30}$	$T_{50}$		
Actual Fault	Baseline	$N_f$	0.964	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.036
		$N_c$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		$P_{s3}$	0.000	0.000	0.786	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.214
		$P_{50}$	0.000	0.000	0.000	0.893	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.107
		$T_{30}$	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	SDQ	$T_{50}$	0.000	0.000	0.000	0.000	0.000	0.857	0.000	0.000	0.000	0.000	0.000	0.000	0.143
		$N_f$	0.964	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.036
		$N_c$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		$P_{s3}$	0.000	0.000	0.786	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.214
		$P_{50}$	0.000	0.000	0.000	0.893	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.107
$T_{30}$	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
$T_{50}$	0.000	0.000	0.000	0.000	0.000	0.857	0.000	0.000	0.000	0.000	0.000	0.000	0.143		

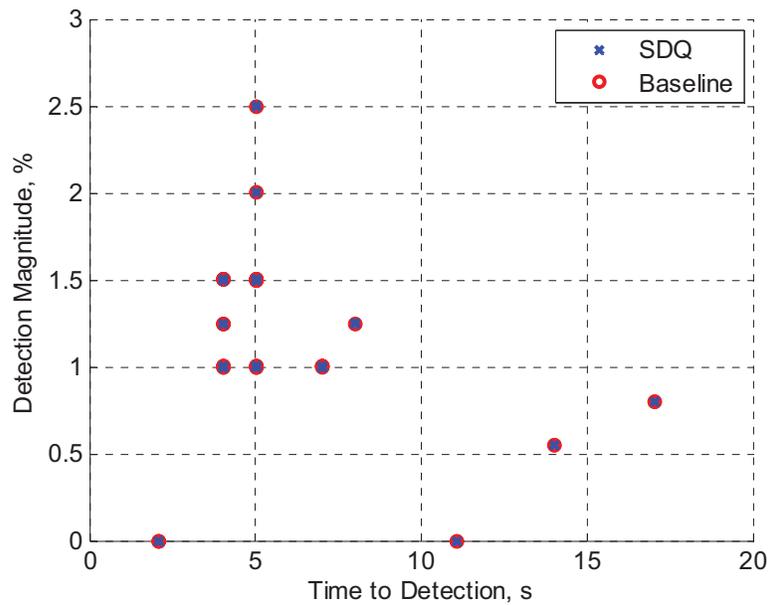


Figure 9.—Plot comparing the  $N_f$  intermittent sensor fault detection magnitude (size of the fault when detected) and the time to detection for both the baseline and SDQ approaches.

To determine if these findings are truly representative for the range of flight conditions that SDQ was designed for, confusion matrices for 26 random takeoff conditions and 26 random cruise conditions comparing the Baseline and SDQ techniques are shown in Tables 9 and 10, respectively. The total number of flight conditions during takeoff and cruise, 26, was arbitrarily selected. For each of the flight conditions, all the sensor faults were simulated and one flight with no faults was simulated to determine if there were any false positives. For all the flight conditions tests, there were no false positives identified (no sensor was disqualified when a fault was not present). The data in Tables 9 and 10 confirm that the results represent the SDQ performance over the broader range of flight conditions, the clear advantage of adding in analytical redundancy by the improved detection rate, and that there is some difficulty in sensing temperature sensor faults compared to rotor speed and pressure sensor faults.

TABLE 9.—CONFUSION MATRIX FOR 26 RANDOM CRUISE CONDITIONS

		Detected and Isolated Faults												No	
		Channel A						Channel B							
		<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>		
Actual Fault	Baseline	<i>Nf</i>	0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.400
		<i>Nc</i>	0.000	0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.400
		<i>Ps3</i>	0.000	0.000	0.503	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.497
		<i>P50</i>	0.000	0.000	0.000	0.514	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.486
		<i>T30</i>	0.000	0.000	0.000	0.000	0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.400
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.514	0.000	0.000	0.000	0.000	0.000	0.000	0.486
	SDQ	<i>Nf</i>	0.890	0.000	0.000	0.000	0.000	0.000	0.027	0.000	0.000	0.000	0.000	0.000	0.082
		<i>Nc</i>	0.000	0.916	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.084
		<i>Ps3</i>	0.000	0.000	0.774	0.000	0.000	0.000	0.000	0.000	0.011	0.000	0.000	0.000	0.215
		<i>P50</i>	0.000	0.000	0.000	0.735	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.265
		<i>T30</i>	0.000	0.000	0.000	0.000	0.742	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.258
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.646	0.000	0.000	0.000	0.000	0.000	0.000	0.354

TABLE 10.—CONFUSION MATRIX FOR 26 RANDOM TAKEOFF CONDITIONS

		Detected and Isolated Faults												No	
		Channel A						Channel B							
		<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>	<i>Nf</i>	<i>Nc</i>	<i>Ps3</i>	<i>P50</i>	<i>T30</i>	<i>T50</i>		
Actual Fault	Baseline	<i>Nf</i>	0.579	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.421
		<i>Nc</i>	0.000	0.608	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.392
		<i>Ps3</i>	0.000	0.000	0.497	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.503
		<i>P50</i>	0.007	0.000	0.000	0.576	0.000	0.000	0.007	0.000	0.000	0.000	0.000	0.000	0.410
		<i>T30</i>	0.000	0.000	0.000	0.000	0.621	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.379
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.536	0.000	0.000	0.000	0.000	0.000	0.000	0.464
	SDQ	<i>Nf</i>	0.880	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.119
		<i>Nc</i>	0.000	0.892	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.108
		<i>Ps3</i>	0.000	0.000	0.730	0.000	0.000	0.000	0.000	0.000	0.129	0.000	0.000	0.000	0.142
		<i>P50</i>	0.007	0.000	0.000	0.763	0.000	0.000	0.007	0.000	0.000	0.078	0.000	0.000	0.146
		<i>T30</i>	0.000	0.000	0.000	0.000	0.737	0.000	0.000	0.000	0.000	0.000	0.145	0.000	0.118
		<i>T50</i>	0.000	0.000	0.000	0.000	0.000	0.691	0.000	0.000	0.000	0.000	0.000	0.158	0.151

## VI. Concluding Remarks

This work focused on applying a previously developed sensor validation strategy, Sensor Data Qualification (SDQ), developed for automated and human in the loop space systems to a commercial aircraft engine simulation, the Commercial Modular Aero-Propulsion System Simulation 40,000 (C-MAPSS40k). The applied SDQ application is a multi-layered approach that contains three different types of checks for diagnosing sensor faults—range checks, rate checks, and analytical redundancy checks applied in the form of sensor networks that capture the relationships between sensors. Range checks protect against a gross sensor fault in which the sensed value exceeds some known minimum or maximum value for the sensor. Rate checks protect against a fast drifting or extremely noisy sensor measurement by checking the derivative of the signal against a known maximum value based on the engine and sensor dynamics. The main feature of the SDQ application was the development of a sensor network intended to detect small magnitude sensor bias drifts of a low rate of change. The advantage of using the sensor network was shown by comparing the results of the SDQ application to a Baseline application which did not feature the sensor network. The sensor network approach excelled for the slow sensor drifts faults and also improved the sensor step change failures but showed no advantage with the intermittent sensor failure which is more dependent on the rate checks. The SDQ application was shown to be able to do an adequate job in detecting faults in rotor speed sensors and pressures, but the ability to detect small magnitude faults in the temperature sensors was shown to be more difficult. Further work could include integrating SDQ with system health management technologies to determine if there is a benefit to an integrated approach.

## References

1. Merrill, W.C., “Sensor Failure Detection for Jet Engines Using Analytical Redundancy,” AIAA 84-1452, AIAA/SAE/ASME 20<sup>th</sup> Joint Propulsion Conference, Cincinnati, OH, June 11-13<sup>th</sup> 1984.
2. Curiac, et al., “Redundancy and Its Applications in Wireless Sensor Networks: A Survey,” WSEAS Transactions on Computers, Issue 4, Vol. 8, April 2009.
3. Merrill, W.C. and DeLaat, J.C., “A Real-Time Simulation Evaluation of an Advanced Detection, Isolation and Accommodation Algorithm for Sensor Failures in Turbine Engines,” A87-13301 03-63, IEEE, American Control Conference, Seattle, WA, June 18-20, 1986.
4. Merrill, W.C., DeLaat, J.C., Kroszkewicz, S.M., and Abdelwahab, M., “Full-Scale Engine Demonstration of an Advanced Sensor Failure Detection, Isolation, and Accommodation Algorithm – Preliminary Results,” AIAA 87-2259, AIAA Guidance, Navigation, and Control Conference, Monterey, CA, August 17-19<sup>th</sup>, 1987.
5. Healy, T.A., Kerr, L.J., and Larkin, L.J., “Model Based Fuzzy Logic Sensor Fault Accommodation,” 97-GT-222, ASME, Journal of Engineering for Gas Turbines and Power, Vol. 120, July 1980.
6. Maul, W.A., Melcher, K.J., Chicatelli, A.K., Sowers, T.S., “Sensor Data Qualification for Autonomous Operation of Space Systems,” AIAA FS-06-07, Proceedings of the American Association for Artificial Intelligence 2006 Fall Symposium on Spacecraft Autonomy, October 13-15, 2006.
7. Melcher, K.J., Fulton, C.E., Maul, W.A., and Sowers, T.S., “Development and Application of a Portable Health Algorithms Test System,” NASA/TM—2007-214840, July 2007.
8. Wong, E., Fulton, C.E., Maul, W.A., and Melcher, K.J., “Sensor Data Qualification System (SDQS) Implementation Study,” NASA/TM—2009-215442, April 2009.
9. May, R.D., Csank, J.T., Lavelle, T.M., Litt, J.S., and Guo, T., “A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller,” AIAA-2010-6630, 46<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Nashville, TN, July 25-28, 2012.
10. Csank, J., May, R.D., Litt, J.S., and Guo, T.-H., “Control Design for a Generic Commercial Aircraft Engine,” AIAA-2010-6629, 46<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Nashville, TN, July 25-28, 2010.





