D1

EVOLUTION OF SHUTTLE AVIONICS REDUNDANCY MANAGEMENT/FAULT TOLERANCE

Jack C. Boykin and Joseph R. Thibodeau
NASA Lyndon B. Johnson Space Center
Houston, Texas

and

Henry E. Schneider
McDonnell-Douglas Technical Services Company
Houston, Texas

ABSTRACT

The challenge of providing redundancy management (RM) and fault tolerance to meet the Shuttle Program requirements of fail operational/fail safe for the avionics systems was complicated by the critical program constraints of weight, cost, and schedule. This paper addresses the basic and sometimes false effectivity of less than pure RM designs. Evolution of the multiple input selection filter (the heart of the RM function) is discussed with emphasis on the subtle interactions of the flight control system that were found to be potentially catastrophic. Several other general RM development problems are discussed, with particular emphasis on the inertial measurement unit RM, indicative of the complexity of managing that three-string system and its critical interfaces with the guidance and control systems.

PROGRAM REQUIREMENTS AFFECTING FAULT TOLERANCE/RM

Space Shuttle Program requirements dictate fault tolerance in all systems other than primary structure, thermal protection, and pressure vessels. In the case of avionics systems, those requirements are specified to be two-fault tolerant, or fail operational/fail safe (FO/FS) (ref. 1) as commonly referred to in Shuttle terms. Considering the design life goals of 100 missions and 10 years or more operational span, this FO/FS requirement not only appears reasonable but almost minimally mandatory. Given a free hand in hardware provisioning, meeting the FO/FS requirement may have been obtainable. As would be expected in a program heavily pressed with cost and weight constraints, the avionics designers' hands were not so free.

Providing FO/FS fault tolerance with no holes or subtle escapes would most easily be accomplished by providing independent five-string operation with independent data inputs to each of the five strings and command functions requiring three out of five votes before execution. This approach would obviously be costly in weight and cost of wiring and avionics components but does offer an FO/FS design approach that could be void of in-flight failure effects concerns for two failures. It would also eliminate the historically complex hardware/software requirements for fault detection and isolation since the first two failures would be transparent. This five-string approach is indeed being implemented in the flight-critical interfaces of the Centaur payloads with the Orbiter. The approach was never seriously considered for Shuttle/Orbiter design because of the obvious systems weight and cost impacts, yet the FO/FS requirement was maintained.

The first step down from the five-string design to meet FO/FS is to provide four-string operation with fault detection and isolation of at least the first fault. Various schemes can be implemented at the four-string level to choose proper data or perform the proper command function, but the susceptibility to the second failure requires a fault detection and isolation scheme to eliminate the first failure before occurrence of the second failure in order to be FO/FS. Two simultaneous or near-simultaneous (within the timespan required to detect and isolate the first failure) failures will defeat this approach; however, Shuttle Program management has granted detection and reconfiguration time in its FO/FS requirement provided the detection is reasonably assured of working. In this scheme, the second failure is then tolerated by selecting the middle valued data of the remaining three inputs and voting two of three for command functions or providing enough muscle in the two good command channels to override the second failure channel. This four-level approach was selected for a major portion of the Shuttle avionics and has fared well with the possible exception of null failures in sensors normally operating about their null, thereby hindering the fault detection of either the first or the second null failure.

The next step down in redundancy while maintaining FO/FS fault tolerance is to provide three-string operation in conjunction with fault detection and isolation of both the first and second failures in the system. Detecting and isolating the first failure in a three-string system is not overly complex and detecting the second failure through two-level comparison of inputs or outputs is straightforward. The difficulty in this approach comes in isolating the second failure. Disagreement between only two inputs requires a decisionmaking vote that is often costly and still not 100 percent

certain, such as built-in test equipment (BITE), self-tests, or reasonableness tests. In the cases where isolation is still lacking, a crew "guesstimate" and manual reconfiguration may be used to supplement the auto RM. This approach can also be costly and leaves room for errors. This second fault isolation problem was a primary factor in the basic four-string avionics design in the Shuttle with the Orbiter inertial measurement units (IMU's) being the most notable exception. The basis for this exception, as with the other three-string systems, included weight and cost, reliability background and history of similar hardware, and projected BITE capabilities to cover second failure isolation. Treatment of the second failure detection and isolation of the IMU's has involved such extensive efforts in analysis, verification, software changes, and flight procedures development that the possibility of adding a fourth IMU to the Orbiter is still under consideration. Indeed, the complexities of timely detection and isolation of two-level IMU faults and the capabilities and limitations of the IMU BITE design are sufficient to warrant treatment as an individual paper, accounting for the large portion of this paper being dedicated to IMU RM. A considerable amount of the IMU RM design change activity deals with fine tuning of the data to minimize errors passed to the guidance and control systems; however, the final fault tolerance assessment accounting for the two-level isolation problems results in no capability to isolate approximately 0.7 percent of the second faults and an even more disconcerting 0.15 percent of the second faults which can result in selection of the failed IMU (ref. 2).

Any further reduction in redundancy level below three strings obviously fails to meet the avionics FO/FS requirement unless, of course, the function itself is not required and is then by definition FS after the second failure. The net result is that Shuttle avionics basically evolved as a three- and four-string design attempting to provide fault detection and isolation capability to make these redundancy levels equal to the five-string design required to be purely FO/FS. The success of this design is not easily measured; however, there are several key data points to recount when considering Shuttle redundancy design versus a blanket FO/FS requirement. First, the Orbiter avionics alone has officially documented some 255 critical-items lists (CIL) exceptions to the FO/FS requirement. Secondly, in striving to meet the blanket FO/FS requirement, the less than pure approach has resulted in an analysis and verification program of staggering proportions. As former astronaut and Orbital Flight Test Manager Donald K. Slayton stated (ref. 3), though possibly too late to influence the Shuttle Program direction, there is an unpredictable but "high cost of worrying improbable possibilities." Because of the second fault tolerance requirement and the fault detection and isolation deficiencies (delays in isolation and/or various degrees of escapes), every combination of failures, however improbable, must be analyzed and verified as acceptable. This is then magnified by reconfiguration actions planned to optimize the postfailure system configuration, each of these obviously requiring verification. In this process, as is its objective, escapes from the FO/FS requirement are sometimes discovered, leading to software, hardware, and/or procedural changes which, of course, add to the verification work. Quantification of such changes is nearly impossible but some understanding of the problem can be evidenced by the ever-present hardware and software change boards, the approximately 700 pages of crew malfunction procedures, the more than 1000 pages of off-nominal crew procedures, the libraries full of off-nominal verification procedures and test reports, and the more than 250 software change requests which have currently been approved to the RM Flight System Software Requirements (FSSR) alone (ref. 4).

Mr. Slayton's approach to the problem was proposed basically as drawing a line in the sand defining some failure criticality not based on a blanket FO/FS requirement but rather based on the probability of that failure (or combination of failures) to occur. If the failure probability is determined to be lower than that line, the design would not have to accommodate changes to meet the FO/FS requirement. The obvious problems of choosing the acceptable failure probability and the mechanization of determining the probability of potential failures are admittedly nontrivial; however, Mr. Slayton cites the success of the Saturn 5 program as a working example of this approach.

An alternate approach that has essentially been described earlier is to provide a five-string operation with data and command selection, based on the middle value of the five or voted to require three of the five data or command statuses to produce a functional response. The aforementioned weight and cost penalties of such a system are understandably undesirable and easily considered unjustified in the preliminary stages of system design. Considering the historically costly change traffic and verification activities that go with the attempted development of less than pure redundancy matching requirements, the initial acceptance of the additional string weight and cost may have been an effective overall decision for the Shuttle/Orbiter Program.

IMPACTS OF SUBSYSTEM PERFORMANCE CHARACTERISTICS ON RM

The Shuttle RM involves both hardware and software functional implementations, the extent of each determined by the design and response characteristics of the individual subsystem element. As with any development program on the level of sophistication and gross technical expansion of functional capabilities such as in the Shuttle Program, the potentials for hardware and software design

deficiencies are not few. Adding the scheduling demands typically forced on NASA programs complicates the situation by requiring parallel development of subsystem hardware and the associated applications and RM software. This parallel RM design does not then allow for an individual line replaceable unit (LRU) performance understanding beyond that established as design requirements in the procurement specifications. Additionally, the benefits of failure modes and effects analyses on the LRU's cannot be utilized in initial designs, since these analyses are not completed sufficiently until final design and/or design verification testing has been completed. Finally, the performance of the LRU's within the system as a whole cannot be adequately anticipated until integrated systems simulations and/or verification programs are underway involving flight hardware/software designs and sophisticated models of environments, aerodynamic effects, and vehicle dynamic models. Development of the RM schemes under these handicaps can and did result in many initial assumptions that turned out to be invalid or at least not sufficient to provide totally acceptable RM.

A primary example of this programmatic dilemma occurred in the Orbiter flight control system (FCS) interface with the redundant FCS sensor systems. Providing the FCS with the best available rate and acceleration data from a quad redundant sensor set at first glance appeared trivial. The program had provided the fourth sensor set after only three had been initially proposed because it was recognized that isolation of the second failure in dynamic periods of flight could not be guaranteed within FCS control limits. The formulation and use of a software-derived fourth input was too complex and laced with its own shortcomings.

The selection of the "best" data input from this set was considered to be a simple extension of the three-sensor method; i.e., provide a software selection filter (SF) which dealt with the first three inputs by simply selecting the middle valued input for use by the FCS, substituting the fourth sensor input only after one of the original three had been determined to be failed (fig. 1). This SF design was baselined and implemented in the software RM along with a fault detection, isolation, and reconfiguration (FDIR) scheme that used a unit-to-unit comparison test to determine whether a unit had failed. On the surface, this approach was simple and foolproof; in final application, it was not so simple and indeed inadequate. Two key conditions combined to defeat this original approach: (1) to eliminate or limit false alarm exposure, the magnitude of the differences between units had to be significantly large enough to account for unfailed unit normal deviations (3σ), system noise, and transients; and (2) the FCS control laws and vehicle responses aimed toward and generally achieved stable flight conditions within the limits of times and/or magnitudes established to prevent the false alarms. The interaction of these two factors resulted in a general inability to detect and isolate a null failed sensor (fig. 2).

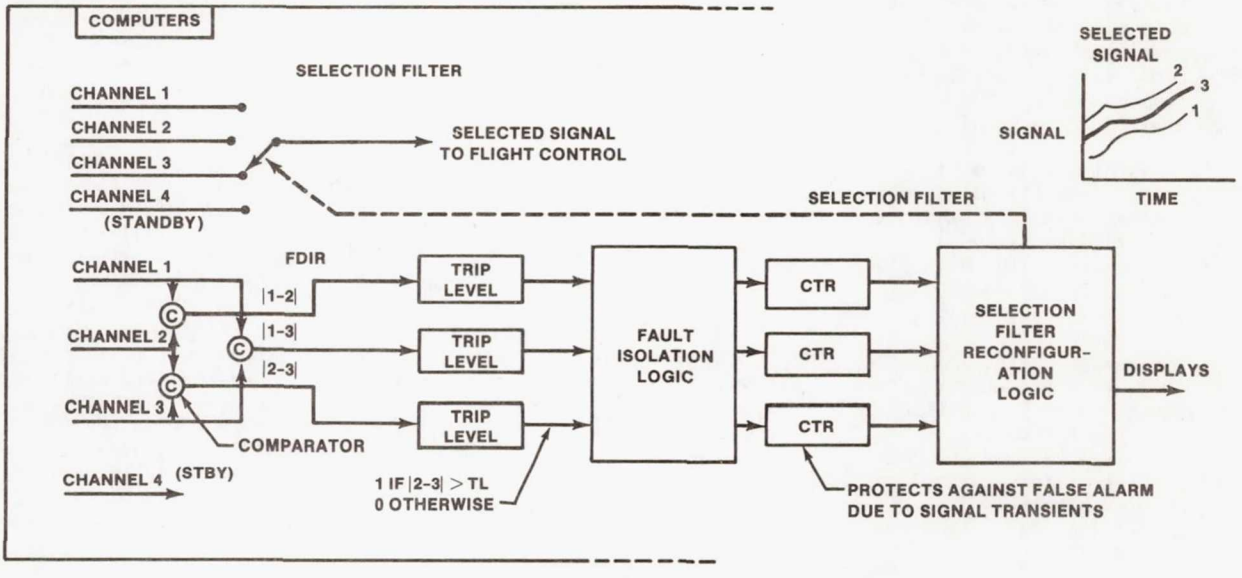

FIGURE 1.- INITIAL SOFTWARE QUAD RM APPROACH.

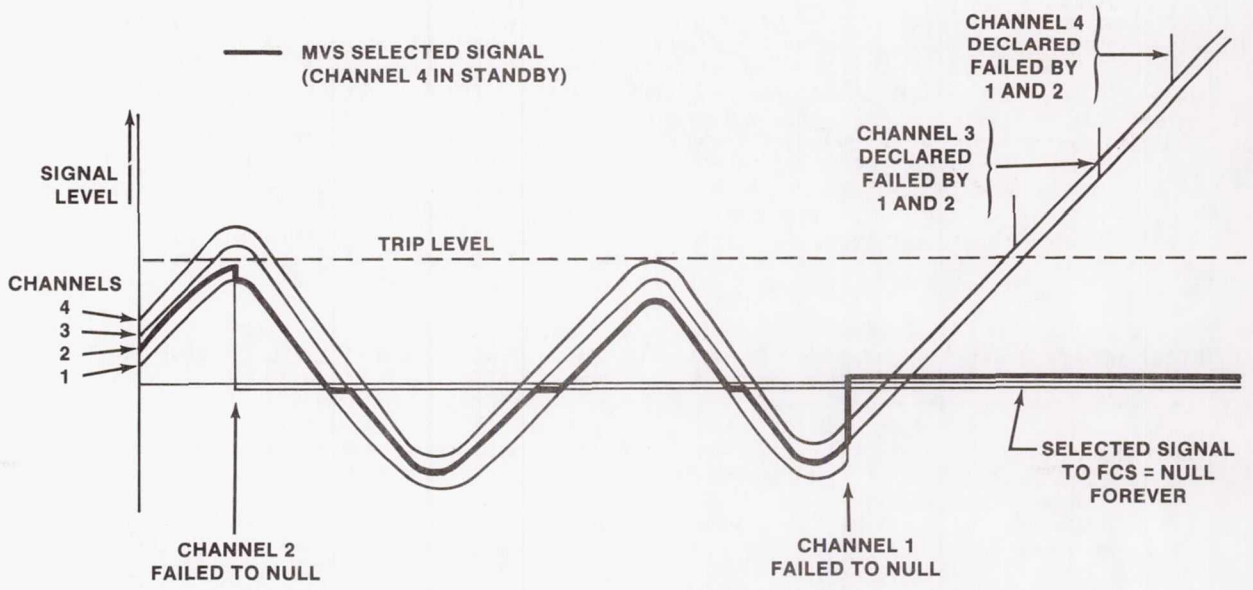

FIGURE 2.- MVS NULL FAILURE SCENARIO.

For the first null failure experienced, this was an insignificant event since the SF still provided a good output in the form of the smaller of the two remaining good units or the null if the two remaining good units were operating near to and bracketing the null failure. With the first null failure included in the SF set, however, the effects of the second null failure became catastrophic. With a second null failure in the set, the middle value in the SF will always obviously be a null. As vehicle stability deviates from this point, the FCS will not see the change and quickly becomes unstable. An additional irony is that the RM will actually declare a failure against the only good sensor input in the SF as its output builds, deselect that sensor, and replace it with the fourth (good) sensor, only to have it defeated just as the first good sensor was. The recognition of the inability of the baselined RM to deal with this dual null scenario was not possible until the hardware and systems models were completed and evaluation and verification test facilities could be used. Verification analyses showed the dual null system effects to be an unstable vehicle in ascent and entry mission phases for the pitch, roll, or yaw rate gyros and severe violation of the load relief and g limits for such loss of the normal or lateral accelerometers.

The risk associated with the dual null deficiency in the RM was considered to be extremely small considering the relatively short period of use of the rate gyros and accelerometers. This could be further minimized by performing stim tests just before lift-off and again just before entry, thereby detecting nulls and allowing proper reconfiguration. The risk was not zero, however, and a proposed fix to obtain the highly desirable FO/FS status which involved a software modification only was developed. The impact of this software change was evaluated as an increase in CPU requirements for the SF from 0.875 to 1.46 percent, an increase in memory requirements for the SF from 75 to 90 words, and a decrease in memory requirements for the FDIR from 229 to 200 words. At this point in the program, the new quad midvalue select (QMVS) SF was accepted as the resolution of the dual null failure concerns for the rate gyros and accelerometers. Since that time, the body flap position feedbacks and the solid rocket booster (SRB) rate-gyro assemblies (RGA's) have also been improved by application of the four active inputs SF (QMVS) in place of the three active with a standby.

For flight control and dynamic response limitations, the QMVS eventually proved to be adequate. However, the verification analyses with dynamic flight models discovered yet another sophisticated escape. The QMVS SF was observed in these verification activities as meeting the flight control stability requirements with dual nulls present. However, an unexpected increase in reaction control system (RCS) jet activity was resulting in significant increases in RCS propellant consumption. Refinement of these cases determined that with the dual null failures and reasonable biases on the remaining two sensors, RCS propellant consumption could increase by as much as a factor of four (ref. 5). The contribution of the QMVS SF to this phenomenon is illustrated in figure 3.

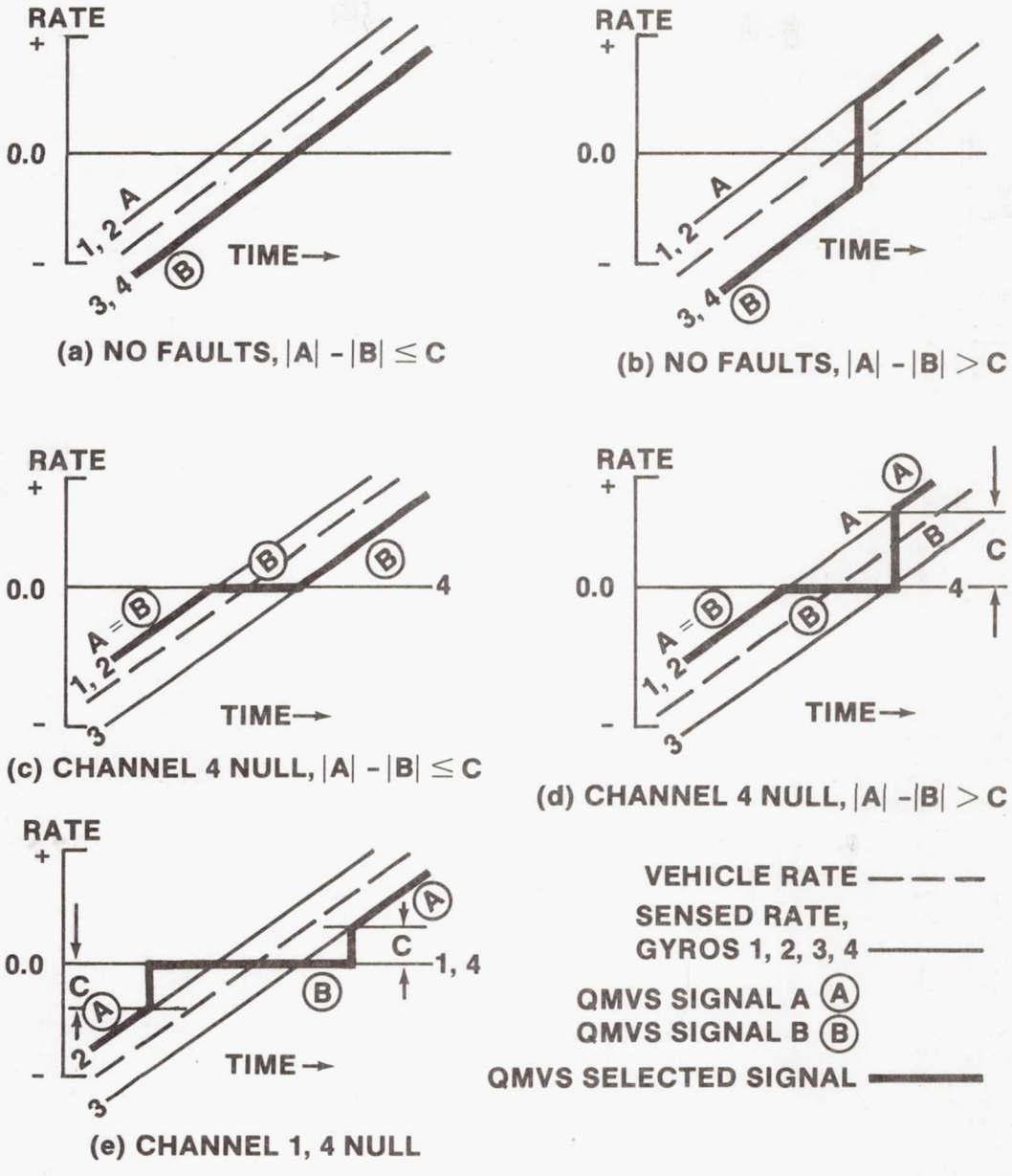

FIGURE 3.- QMVS SELECTION FILTER OUTPUT COMPARED TO VEHICLE RATE AND SENSED RATE.

The QMVS basically works as illustrated in figures 3(a) and 3(b) in that the SF logic evaluates all four inputs, determines the two middle valued inputs, and selects one of them based on a test of their difference between each other (the "C" value). The "C" value is fixed based on the reasonably expected null offsets between good gyros, null offsets between good multiplexer/demultiplexer (MDM) channels, and MDM nonlinearity. It is designed to prevent discontinuous SF outputs caused by switching between signal A and signal B for the no-fault case and yet provide satisfactory rate outputs with undetected single or dual null failures. Figures 3(c), 3(d), and 3(e) illustrate the QMVS performance with single and dual undetected null features. Because of the fine-tuned FCS and vehicle responses and the typical operation at near null vehicle rates, the illustrated nonlinearities in the sensed vehicle rates can result in residual oscillations and attendant RCS propellant consumption as previously described. Evaluation test cases indicated that under the dual null conditions, sufficient propellant consumption could result in depletion and loss of control unless the condition was recognized and crew intervention was timely.

As another escape from the FO/FS design requirement, however limited in probability, changes were developed as candidate solutions and evaluated in flight performance simulators to eliminate this last caveat from the FCS RM verification status. With the weakness of the QMVS being the deficiency of the FDIR to detect and isolate null failures in the Shuttle FCS environment, one obvious approach was to bias the SF to choose the more active midvalue parameter and stay with it until fault conditions indicated that the other midvalue parameter was significantly more active. The term used to describe this SF approach is interchangeable midvalue selection (IMVS). The IMVS operation under nominal, single, and dual null failure conditions is presented in figure 4. The primary difference in the IMVS and the QMVS is that once the two middle valued parameters of the four-parameter set have been determined (identically in both approaches), the SF chooses the largest of these two values and sticks with that selection until fault conditions are detected and then switches one time only to the other midvalued parameter. Remember that the QMVS selected the highest of the middle valued parameters for dispersions greater than a present value "C" and the smaller middle value for dispersions less than "C," resulting in undesirable discontinuities being provided to the SF user. As previously described, the discontinuities and nonlinearities of the QMVS SF in the presence of dual undetected null failures results in unacceptable propellant consumption. As shown in figure 4, the IMVS eliminates these discontinuities and reduces the nonlinearities, thereby improving FCS efficiency. Evaluation test cases showed propellant consumption reduction compared to the QMVS ranging from 62 to 1100 pounds, depending on the axis containing the null faults, the point of fault insertion, and the biases assigned to the remaining two good parameters.

The IMVS has been approved for Shuttle implementation, providing resolution to the current RM design caveat. It will likewise serve as a verified approach in future four-string design activities. One can only wonder again whether the costs of the changes from MVS to QMVS to IMVS and the associated analysis and verification activities might have compared to initial design implementation of a five-string system.

FAULT TOLERANCE ASSESSMENT OF INTEGRATED SYSTEMS

It has already been pointed out that design of the Shuttle avionics systems to the FO/FS program requirement under the weight and cost constraints which drove designers to not demand the pure five-string approach is at best a complex and difficult task. Several of the major exceptions to full-up redundancy in the Shuttle avionics interfaces are especially noteworthy, specifically the universal servicing systems such as electrical power distribution, cooling, and instrumentation. Of these, possibly the key factor in concern for integrated systems failure tolerance turned out to be the three-string electrical power distribution system. Obviously, with only three sources of power to spread to up to four user interfaces, cross-strapping of power to some or all of the components in critical functions was a necessity. Providing the visibility and assessment of the effects of this and other cross-strapping proved to be the weakness of the avionic fault tolerance assessment.

The typical programmatic tools to assess failure modes and effects were employed, including individual subsystem analyses and some level of integrated systems fault testing in facilities such as the Flight Systems Laboratory (FSL) and the Shuttle Avionics Integration Laboratory (SAIL). Some fault tolerance escapes were discovered through the subsystem analysis, and test programs periodically resulted in unexpected problems under fault or off-nominal conditions which could be catastrophic if occurring in flight. These escapes, though almost all attributable to the less than total redundancy design, were basically the result of the overall system complexity and subtleties, which could hardly be expected to be foreseen by the individual subsystem designer performing his failure modes and effects analysis (FMEA). Some of these factors include understanding the timing of the failures with respect to each other and to the mission phase, understanding the software mechanization and its dependence on time homogeneous data, understanding the normal and malfunction procedures which might be used by the crew to operate the systems and control configuration, and finally, understanding the individual subsystem functional impacts caused by failures in the universal service subsystems.

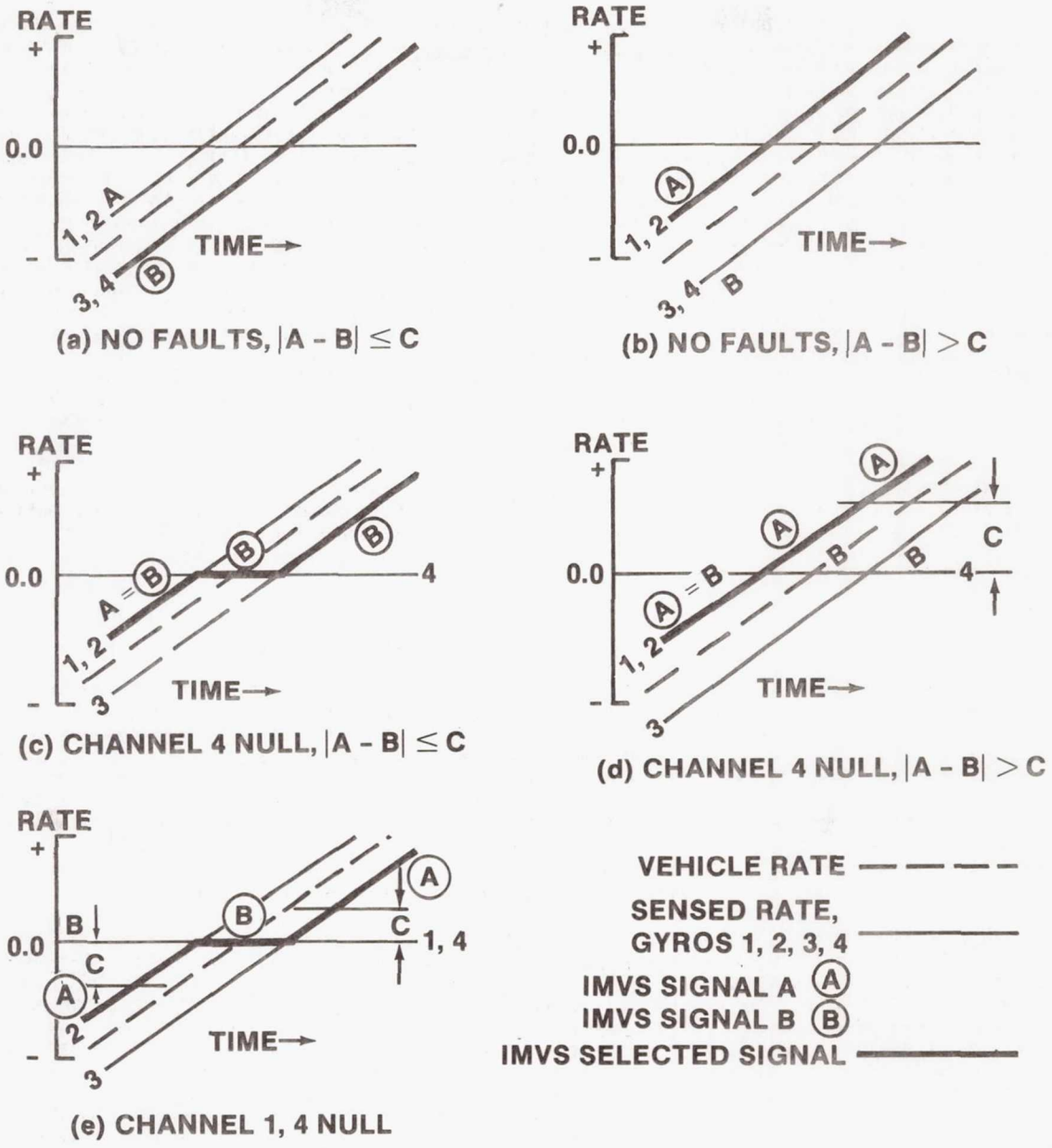

FIGURE 4.- IMVS SELECTION FILTER OUTPUT COMPARED TO VEHICLE RATE AND SENSED RATE.

This integrated fault tolerance assessment escape mechanism can be demonstrated by the example in figure 5. The FCS muscle for control during early entry stages is provided by the aft RCS jets. These jets, unlike other flight control effectors, employ single-string authority for jet firing, relying on the quantity of jets to satisfy the FO/FS requirement. The FCS avionic interface to fire these jets is provided by the reaction jet driver (RJD) circuits, which are divided into four channels with each channel receiving dual power inputs from the three-string EPD system such that any two failures can disable only two of the four RJD channels. The instrumentation monitoring of the RJD/jet system is provided in only two signal conditioning units; however, each one contains isolated internal modules and each box receives dual power inputs from the three-string EPD system such that any two failures can disable only half of the instrumentation system. In like manner, the data management/command interface is a four-channel system with identical redundancy to the serviced RJD's (including power redundancy). Taken piece by piece, the flight-critical function of RCS jet control appears to be FO/FS and each subsystem-level FMEA would support that conclusion. On closer inspection from an integrated, functional end-to-end viewpoint, with software performance considered, it becomes obvious

that the fault tolerance is not completely FO/FS, because of the channelization of the EPD inputs to each subsystem element. As can be seen in figure 5, if the aft local power buses A and C were lost, the total power to half of the RCS control circuitry would be lost (RJD's 1A and 2A), along with their respective data management interfaces (MDM's FA3 and FA4). Through cross-strapping, the remaining half of the RCS jets would be normally sufficient to control the vehicle. The total system reaction to the loss of buses A and C, however, includes loss of the instrumentation subsystem's signal conditioner OL2, which happens to provide service to the RCS jets controlled by RJD 2B, resulting in unresponsive data monitoring of those jets. The RM software will respond to this situation by recognizing the communications fault in strings 3 and 4 and suspending processing of those strings. The signal conditioning fault in string 2, however, is not discernible from actual low value data response to the RM, thereby leading to a failed "leaking" and/or failed "OFF" conclusion by the RM. Either of these events results in a deselection of the affected jets, with the final result being that the FCS muscle for the two bus failures is cut to a single set of jets which cannot maintain vehicle control.

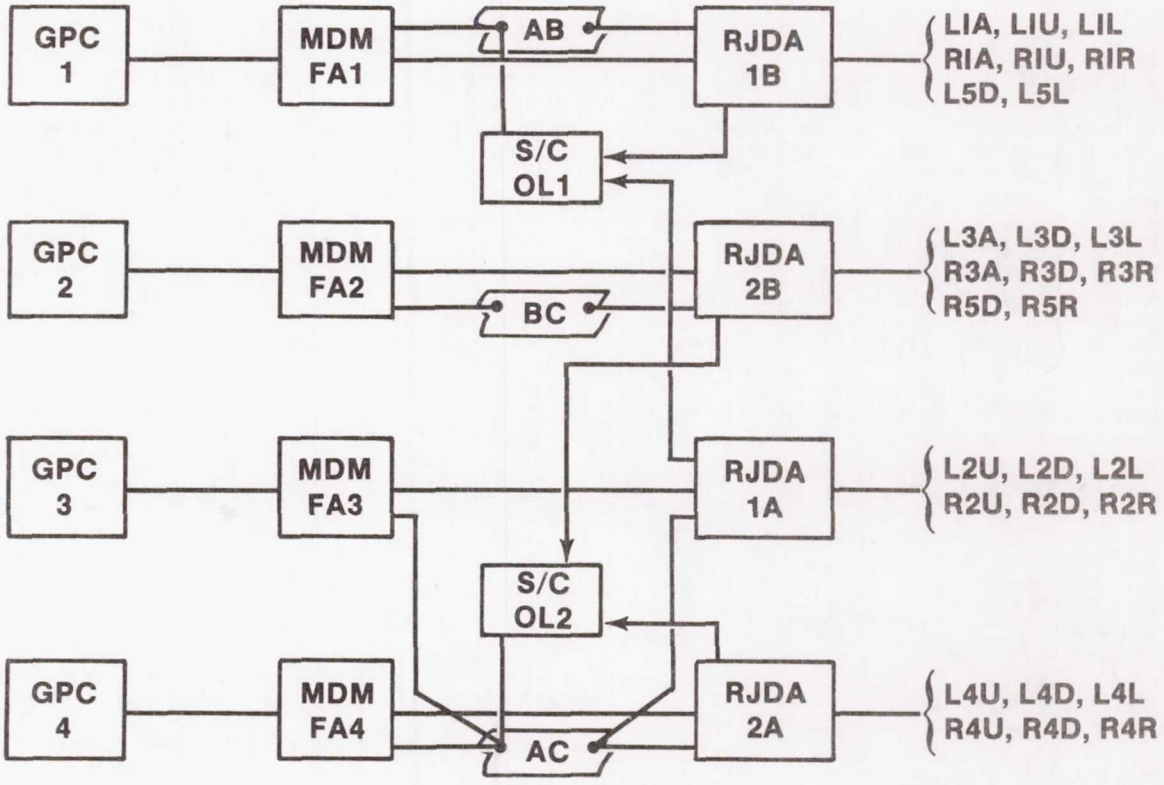

FIGURE 5.- AFT RCS CHANNELIZATION ESCAPE.

The loss of two power buses in a basic three-bus system is obviously a low-probability situation. It is, however, required to be addressed in assessment of an FO/FS requirement and is the reason for the multiple cross-strapping schemes on the Shuttle. As this and other specific escapes from the FO/FS requirement were discovered in test facilities and flight environments with failures actually present, the potential for further unknown escapes in a system as complex as the Shuttle hardware/software system became painfully obvious. The limitations of resources to perform actual verification tests for every combination of failures, under every critical mission phase, with every version of hardware and software to be flown, with every reasonable crew procedural response, and with every reasonable variation of flight environments are equally as painful. Recognizing this problem, the Shuttle Program established an "Avionics Audit" task to "perform a study and analysis that

identifies the fault tolerance capability of the integrated Orbiter avionics with respect to functional authority or influence derived from hardware channelization, hardware cross-strapping, software cross-strapping, control points, crew procedures, and external environments."[1] This ongoing task has been funded for more than $1 000 000 directly with unestimable program costs associated with study management and review and hardware and software changes resulting to date.

As previously implied, FO/FS implementation through a five-string approach would architecturally eliminate essentially all of this type of escape. Lacking that luxury, programs as complex as the Shuttle avionics must establish high-priority activities, such as FMEA's and sneak circuit analyses, and integrated end-to-end functional fault analysis, such as the avionics audit, early in the design phases of the program to develop a confident position from which to claim conformance with that program's fault-tolerance requirements.

OTHER LESSONS LEARNED IN RM EVOLUTION

Similar to the painful evolution experienced with the RM SF interaction with the FCS performance, a major area of potentially avoidable RM change activity resulted because of hardware performance definitions not being mature early enough to allow adequate RM design. A primary example of this problem is the RCS RM. Propellant leaks in an RCS jet can result in hazardous operation, thereby requiring fault monitoring and reconfiguration by RM to preclude further use of a leaking jet. RCS hardware design provided for a temperature measurement on the oxidizer and fuel valves on each jet and an initially proposed RM algorithm which would declare a jet as leaking when either temperature fell below 48° F. Vacuum chamber testing of actual hardware later provided a better definition of the jet thermal responses to various leak conditions and allowed the RM thresholds to be adjusted in software to 30° F to minimize false alarms and still provide adequate hazard protection. Flight experiences in the early STS missions provided the final environmental operation assessment, which revealed still another hardware performance sensitivity and required further RM software adjustments. It was determined that the upfiring primary jets experienced a significant cooling effect in the early stages of entry where the "g" field holds the postfiring "dribble" propellant in the jet and the near-vacuum environment enhances the cold-inducing evaporation/sublimation of that propellant. This phenomenon actually resulted in several false alarm/deselects on early missions where jets were relatively cool before reentry, resulting in a software change to provide an even lower fault detection limit and minimize the false alarms. Another erroneous original assumption which guided RCS RM design was that the large primary jets would respond to leaks in the same fashion as the smaller vernier jets. Again, vacuum testing of actual hardware under leak conditions provided the surprise that the vernier jets could not leak enough to provide for sufficient cooling effects to ever reach the limits established for the primary jets. The resulting RM change to account for this hardware performance characteristic included RM software mods, heater thermostat setting changes, crew procedural impacts, and flight usage restrictions based on vernier jet warmup sufficient to ensure RM leak protection. The problems evidenced by experiences such as these, although not totally avoidable, do point out the need to develop as completely as possible the hardware performance characteristics under true flight conditions before fixing the associated RM schemes. Lacking that luxury, modification impact could be minimized by implementation schemes which include as much flexibility as can be afforded, such as separate I-loadable limits for each data input to an algorithm.

A second area of continuing RM design modification and assessment is the dilemma of providing the best fault protection thresholds possible while not allowing any significant chance of a false alarm. A key factor in this dilemma is the 3σ program-established variation to be allowed in hardware and system performance. This implies that RM should not declare an LRU failed that is within 3σ of the "normal" performance, and further that, since -3σ is as good as +3σ, actual differences between usable LRU's must allow for somewhat more than 3σ (statistically reasonable to assume $\sqrt{2}$ x 3σ). Accounting for this type of variation and arbitrarily selecting a design goal to have no more than one false alarm in 500 missions (the approximate original number of missions in the Shuttle Program), the fault detection thresholds can be significantly large so as to mask some LRU failures and/or result in system errors or transients that can jeopardize mission performance. Determination of the smallest acceptable thresholds to minimize false alarms while providing adequate fault detection is another late-blooming product due to the necessity to have statistically accurate performance assessments. This involves not only certifiable LRU standard deviation data but also development of accurate vehicle systems and flight profile environments. Although this development task is not easily hurried or readily avoidable, some potential areas for minimizing the dilemma do exist. Premium dollars in the LRU design and production phase could result in the guaranteed 3σ variation of some LRU's being significantly smaller. Indeed, the exact same hardware design could sometimes be

---

[1]Contract Change Authorizations 892 and 1032 to NASA Contract 9-14000, Schedule A.

"improved upon" by simply paying for screening and documentation of its quality. A variation of this theme could be to establish a set of thresholds based on periodic monitoring and verification of the actual variation of the flight hardware. One-sigma deviations are considerably more normal in the Shuttle hardware performance to date, and thresholds based on more realistic variations could obviously be tighter so as to improve system performance while maintaining the same realistic false alarm sensitivity. The final solution is, of course, to provide five-string operation as previously described, thereby eliminating the requirement for fault detection and the dilemma of protection thresholds versus false alarms.

EVOLUTION OF SHUTTLE AVIONICS FAULT TOLERANCE - LESSONS LEARNED

IMU REDUNDANCY MANAGEMENT

The lessons learned for IMU RM have occurred mainly in three areas: (1) understanding the hardware and its failure modes and error characteristics; (2) understanding the design of the software and its ability to attenuate errors and protect against transients; and (3) making the software conformable to variations in mission plans, flight rules, or crew procedures. Key facets of these problem areas are summarized in table 1.

TABLE 1.- LESSONS LEARNED - IMU REDUNDANCY MANAGEMENT OVERVIEW

Area	Problem
Hardware	• Error modeling, failure modes • Warmup transients, trending, aging • Heading sensitivity • Single-point failures • BITE use, performance, sensitivity

Software	• Threshold formulation, queuing, resetting
	• Dilemma resolution, lack of FO/FS
	  • IMU redundant gyro BITE use
	  • Strapdown rate-gyro use
	• Transient protection, velocity and attitude selection filter design
	  • Velocity downmode transients
	  • Attitude dithering

Flight operations	• Extended launch holds
	• Early alinement/delayed entry
	• Computer/LRU reconfiguration
	  • Freeze-dried GPC
	  • Commfaulted LRU's

Three gimbaled IMU's supply velocity and attitude data to the airborne computers, which have the ability to automatically detect and isolate up to two mission-critical failures. Setting thresholds as bounds for normal or acceptable performance has been one of the critical software design tasks. Setting thresholds below the envelope of normal measurement error can lead to nuisance false alarms and premature loss of an IMU. Allowing for normal or expected divergence in measurement error can lead to transients when changing from the use of one IMU to another, as well as to degraded system performance before isolation of the second failure.

To avoid or minimize these difficulties, selection filters (operating at either 6.25 or 1 hertz, according to need) are used to protect against hardover failures, to control switching transients, and to inhibit or attenuate error growth before isolation of the second failure. Tracking tests are used for failure detection and isolation (FDI) at a background rate of 0.0625 hertz to protect against soft performance failures. To guard against a wrong identification in the event of a transient error condition, an IMU can be permanently failed only after a number (n-count) of successive identical fault identifications occur in consecutive passes of the fault identification logic.

USE OF BITE

Without adequate failure-to-noise margins, it is unreasonable to expect failure isolation by means of software tests. To avoid this difficulty, BITE is used as the first-priority discrimination after detection of a failure in a pair of IMU's. If BITE does not discriminate, the software isolation tests are used to identify the failed IMU. Use of BITE in this manner avoids the possibility of BITE false alarms causing premature loss of an IMU whose attitude and velocity measurements continue to be good. Also, software tests are vulnerable to wrong isolation decisions when exposed to simultaneous multiaxis failures - failures of the kind most often isolatable by BITE (a tumbling platform, for example).

If a fault remains unidentified after a given number of consecutive fault detections, an IMU dilemma is signaled to the crew to indicate the need for manual intervention.

FAILURE MODES

The software design was conceived to detect and identify soft bias shifts in attitude or velocity outputs. One of the great fears was to falsely detect errors and downmode good units, which was recognized as an easy way to catastrophe. Part of this fear was also detection, identification, and reconfiguration which could occur for a transient error condition which was conceivable for a lightning strike or temporary communications failures in a data bus link between the IMU's and the general-purpose computers (GPC's), hence the use of an n-counter logic.

These concerns basically have led to a design that is tolerant of system noise. During STS-6, 5000μg noise was evident on the IMU 3 z-axis accelerometer during the entire flight. The noise was as high as 30 000μg, evidenced by the velocity underlimit (VUL) BITE alarms during entry. This IMU was a good navigator and was used almost exclusively for attitude reference during STS-6.

Because the noise was contained only on the one z-axis accelerometer, IMU 3 would have been most likely deselected for a threshold violation. If the noise were correlated to other axes, the resulting error could resemble a simultaneous multiaxis failure. The danger here is that, for the two-level IMU case, a unit could be deselected based on random error with a small but real risk that a good unit could be removed and the noisy unit could remain in candidacy. Although an observant crew could manually deselect the offending unit, procedures for trapping a noisy unit are being investigated.

SOFTWARE

Although we had confidence software could be designed to handle failures in two IMU's and obtain FO/FS performance in three IMU's, the coverage of failure rate is not 100 percent in a pair of IMU's. The BITE coverage is on the order of 90 percent of the failure rate. Of the remaining 10 percent, the software coverage is on the order of 60 to 80 percent at best, giving a total coverage of 96 to 98 percent of the failure rate for the hardware/software system. Although, in theory, we could achieve 99.8 percent of the mission-critical failure rate of 300.4×10^{-6} failures per operating hour, operational considerations such as extended holds or delayed entry can reduce coverage.

Lack of 100-percent FO/FS coverage in IMU RM has led to reliance on other available sensors especially for attitude during entry where a two-case dilemma could be catastrophic for a hard-failure condition. A body-mounted RGA supplying instantaneous body rate to flight control is used to integrate and maintain an inertial body-attitude state during a two-case IMU attitude dilemma, thus creating an artificial fourth IMU for attitude. The maintenance of the integrated fourth IMU attitude state from RGA data does not result in an inertial quality attitude reference. Although the RGA's produce a good instantaneous body rate, noise and quantization result in an inferior integrated attitude state (e.g., approximately 70^{o} per hour effective drift rate). The addition of a fourth IMU could eliminate the risk of the IMU RM system defaulting to the RGA's forever dilemma as well as simplify the design of the IMU attitude selection filter and attitude processor.

Differentiating IMU attitude data to supply a quality body rate to flight control could potentially eliminate the need for the rate-gyro assembly and the complexities of selection filtering of the body rate data. Refinements in the software area include integrating and maintaining a separate quaternion of body attitude from each IMU for flight control. Quaternion averaging could eliminate attitude dithering between prime selected IMU's. Averaging could also provide performance enhancement and open up the tight margins that now exist between mission performance requirement and instrument capability.

Because performance of the guidance, navigation, and control (GN&C) system is degraded when errors escape detection, the propagation of second failure errors has received considerable attention. Everyone is aware that the effects of a first failure in three IMU's can be avoided. Everyone

knows performance degradation cannot easily be avoided when one IMU in a pair fails. Although prime selecting a unit in a pair for measurement can avoid ill effect in half the failure cases, there exists the intolerable risk of having selected a failing unit and having to absorb the full impact of a failure in the GN&C system. Averaging measurements in a pair can cut the error exposure in half between the time of inception of a failure and the time of final system reconfiguration. For threshold violations, immediate deselection of an element of the pair based on BITE status can further reduce failure effects, leaving a residual problem of error propagation for a failed unit tracking just below the threshold. To do a good job, the RM designer must be given an error allowance or margin between normal measurement error and system performance requirements. The alternatives are to accept the degraded performance or to have more IMU's.

The lesson learned here is that FO/FS in three strings requires tightening up on instrument error specifications so that, even in the degraded mode before isolation of the second failure, the worst performance is still within the acceptable system performance envelope. This lesson would be an important consideration for an autonomous onboard navigation system where three or four high-accuracy IMU's could limit error without the need for outside manual intervention by the crew or flight controllers.

One-hundred-percent FO/FS probably cannot be achieved in three IMU's. The problem with three strings is after losing one, there are only two left. No matter how much thought is given to failure, a new, unanticipated and unmodeled failure lurks around the next corner. That failure is the next two-string dilemma: the next dragon to be slain.

THRESHOLD FORMULATION

In most instances, values for the isolation threshold are set at 20 percent above the noise envelope generated by the McDonnell-Douglas Monte Carlo studies. The detection thresholds are set at the highest value which still protects the GN&C system from IMU errors that cause violation of mission ground rules, operational constraints, or vehicle safety limits. Sensitivity studies by Rockwell International-Downey were used to set the detection thresholds.

ASCENT THRESHOLDS

The onboard IMU RM software in combination with the IMU hardware BITE will detect and identify the first IMU failure during ascent. It will allow up to a 50-ft/sec velocity error at main engine cutoff (MECO) before detection of a second IMU failure but it may not be able to identify the failed IMU.

Ground-based flight control personnel will be prepared to identify a second failure by comparing the IMU navigation states against the navigation state derived from ground-based radar tracking data. An onboard state vector update is planned during ascent if the predicted onboard velocity error should violate a 40-ft/sec MECO constraint, which is the orbital maneuvering system (OMS) fuel budget for an abort-to-orbit in event of a MECO underspeed.

During ascent, there is no margin between detection requirements and instrument performance at the two-level. An adequate margin for 100-percent probability of isolation of a second failure requires a ratio of 2 for VCONS2/VCONS6 for optimal skewing. The ratio is only slightly greater than one for the I-load values. Onboard dilemmas will occur for accelerometer failures between 5000μg and 16 500μg.

A mean acceleration error of 6000μg will violate the 50-ft/sec MECO velocity accuracy targeting constraint for jettison of the external tank. Delta-V averaging at the two-level will attenuate a velocity bias by one-half so the range of failure dilemmas that potentially must be isolatable by ground-based personnel is from 12 000μg to 16 500μg.

ON-ORBIT THRESHOLDS

Normal instrument performance for platform alinement and drift will not support mission accuracy requirements without frequent IMU realinements, yet there is a necessity to allow extended periods for the crew to sleep. The I-load values for the on-orbit attitude detection thresholds are set to allow a 10-hour sleep period before potential false alarms can occur at the three-level (8 hours for two IMU's). The detection threshold will ramp above the attitude requirement for a safe entry 2.5 hours after alinement. After this time, ground personnel must be prepared to call for a realinement (if the actual IMU attitude divergence should exceed the attitude error constraint for a safe entry) to maintain capability for a safe return in event of a contingency deorbit.

For a protected entry, IMU realinement must occur no earlier than 2 hours before entry interface or 2.5 hours before TACAN acquisition. Earlier alinements result in loss of second failure coverage during entry.

The 100-ft/sec altitude rate error requirement at TACAN acquisition requires that velocity tilt error transmitted to navigation be no more than 1350 arc-seconds during entry. The two-IMU detection threshold ceiling (VCONS8) is set to protect this requirement.

ENTRY THRESHOLDS

IMU RM can detect and isolate both a first and a second IMU failure during entry, allowing errors up to 100 ft/sec in altitude rate at TACAN acquisition for the second failure with one notable exception: if the second IMU failure is a dual-axis failure in the ambiguous direction and not covered by BITE, then there is potential for dilemma at the two-level.

The velocity isolation threshold (VRAMP2) is queued by a manual operation by the crew. The value of VRAMP2 assumes the crew will PRO to MM 304 5 minutes before entry interface (EI). If the crew delays the PRO until EI, the 20-percent margin over instrument noise disappears. If the PRO occurs later than EI, there is a danger of false isolation during entry.

The altitude rate uncertainty should not exceed 100 ft/sec during blackout. The 3σ altitude rate error due to navigation, atmosphere, and winds is 48 ft/sec. The RM community chose to allow approximately 85 ft/sec error in altitude rate due to IMU failures. There are two ways to look at this error:

1. The budget is the requirement minus 3σ Nav in RSS sense.

   $$\dot{h} = \left[100^2 - 48^2\right]^{1/2} = 87 \text{ ft/sec}$$

2. The budget is the requirement minus 1σ Nav in an additive sense.

   $$\dot{h} = \left[100 - 16\right] = 84 \text{ ft/sec}$$

IMU RM LAUNCH HOLD CONSTRAINTS

A greater launch hold capability is needed to avoid potential launch delays resulting from IMU alinement accuracy limitations. Present baseline flight software and I-loads were designed to support a safe entry for an abort once around (AOA) given a 3σ vendor specification IMU for which the preflight gyrocompass alinement was completed 50 minutes before lift-off.

Present flight rules result in a nominal countdown time line with 30 minutes between completion of the alinement and lift-off. The nominal time line provides a 10-minute planned hold at T-20 (just before the OPS 9/1 transition). This hold does not affect the IMU's because the preflight alinement programs are scheduled so that the alinement process continues during the hold. Platform release occurs at the OPS 9/1 transition at T-20 minutes in the count. A second planned hold of 10 minutes is scheduled at T-9 minutes, yielding the 30 minutes between platform release and lift-off.

For the case of 3σ IMU's, the Shuttle flight software IMU RM I-loads allow a maximum time of 110 minutes between platform release and lift-off. Because 30 minutes of this time is spent in the nominal count, 80 minutes are available for unplanned holds. Because flight IMU's are rarely 3σ and their relative performance can be monitored by flight controllers, much longer holds may be possible based on the real-time prelaunch performance. Up to 4 hours may be available between platform release and lift-off based on past performance.

In any event, beginning 100 minutes from platform release, flight controllers are required to identify all IMU failures at the two-level because the flight software no longer has this capability. Second failure identification capability is not restored again until the first on-orbit alinement.

Platform misalinement can be predicted analytically based on expected initial misalinement and nominal gyro drift at platform release:

Equation (1) Up error = RSS (60, 0.02915 T_1) arc-seconds

Equation (2) North or West error = RSS (20, 0.0094 T_2) arc-seconds

where T_1 is time in seconds from completion of gyrocompassing and T_2 is the time in seconds from completion of the VEL_TILT subroutine.

Here, the North, West, and Up coordinates are topodetic and are frozen at the instant of platform release. This epoch North, West, Up frame is labeled (N_0, W_0, U_0). After platform release, the true topodetic North, West, Up frame (N, W, U) will be rotating at Earth rate compared to the inertial IMU frame or the epoch (N_0, W_0, U_0) frame. To find the expected IMU misalinement about North, West, Up at lift-off, it is first necessary to calculate the expected IMU error in N_0, W_0, U_0 coordinates and then to transform this result to the N, W, U frame at lift-off. Table 2 shows the final launch misalinement as transformed for the effect of Earth rotation rate. The earth rotation rate carries the vertical axis error into the West axis. (For a level IMU on the Equator, the vertical axis would point West 6 hours after platform release.) Knowledge of platform misalinement at lift-off allows the use of nominal dispersion analysis sensitivities for predicting navigation state error at MECO or at points of interest along the AOA reference mission profile.

TABLE 2.- IMU MISALINEMENT AT LAUNCH FOR INCREASING HOLD TIME IN OPS 101 (1σ)

Time		North, sec	West, sec	Up, sec	RSS, sec	LEVEL, sec
Hr	Min					
0	0	20	20	63	69	28
	20	24	28	79	86	37
	40	33	43	99	113	54
1	60	45	64	124	146	78
	80	61	88	147	182	107
	100	79	117	167	219	141
2	120	99	148	184	256	178
	140	120	179	196	292	216
	160	145	213	205	330	258
3	180	172	247	209	366	301
	200	203	381	208	405	347
	220	236	314	204	443	393
4	240	270	348	194	481	440

The analytically derived IMU misalinement can be used to predict the time of violation of the IMU RM attitude threshold, thereby estimating the available unplanned hold time. Table 2 shows the misalinement of a single 1σ IMU for the first 4 hours after completion of the preflight alinement. The table shows the misalinement about North, West, and Up, the RSS of the three misalinements (labeled RSS), and the RSS of the two-level errors (LEVEL). At 2 hours, the RSS misalinement is 256 arc-seconds 1σ. For a pairwise IMU tracking test, the parity equation residual would be expected to be $\sqrt{2}$ x 256 or 362 arc-seconds 1σ. The 3σ misalinement would be expected to be 1086 arc-seconds.

The two-level attitude fault detection threshold is 1138 arc-seconds as defined in the current flight software I-loads. (The threshold half-angle of 2.76 x 10^{-3} radian is stored in ACONS5, MSID V97U4215C.) A 3σ IMU pair would be expected to violate the two-level fault detection threshold just beyond 2 hours from platform release. These data are plotted in figure 6, which shows the expected pairwise tracking test errors for 1σ, 2σ, and 3σ IMU pairs compared to the two-level attitude fault detection threshold. A 3σ IMU pair would intercept the threshold 126 minutes after platform release.

It should be noted that IMU misalinements are not completely observable by ground monitoring. Flight controllers observe total relative misalinement between pairs of IMU's and azimuth misalinement is not observable by ground navigation performance analysis. Hence, launch hold performance constraints must be based in part on results of premission simulations.

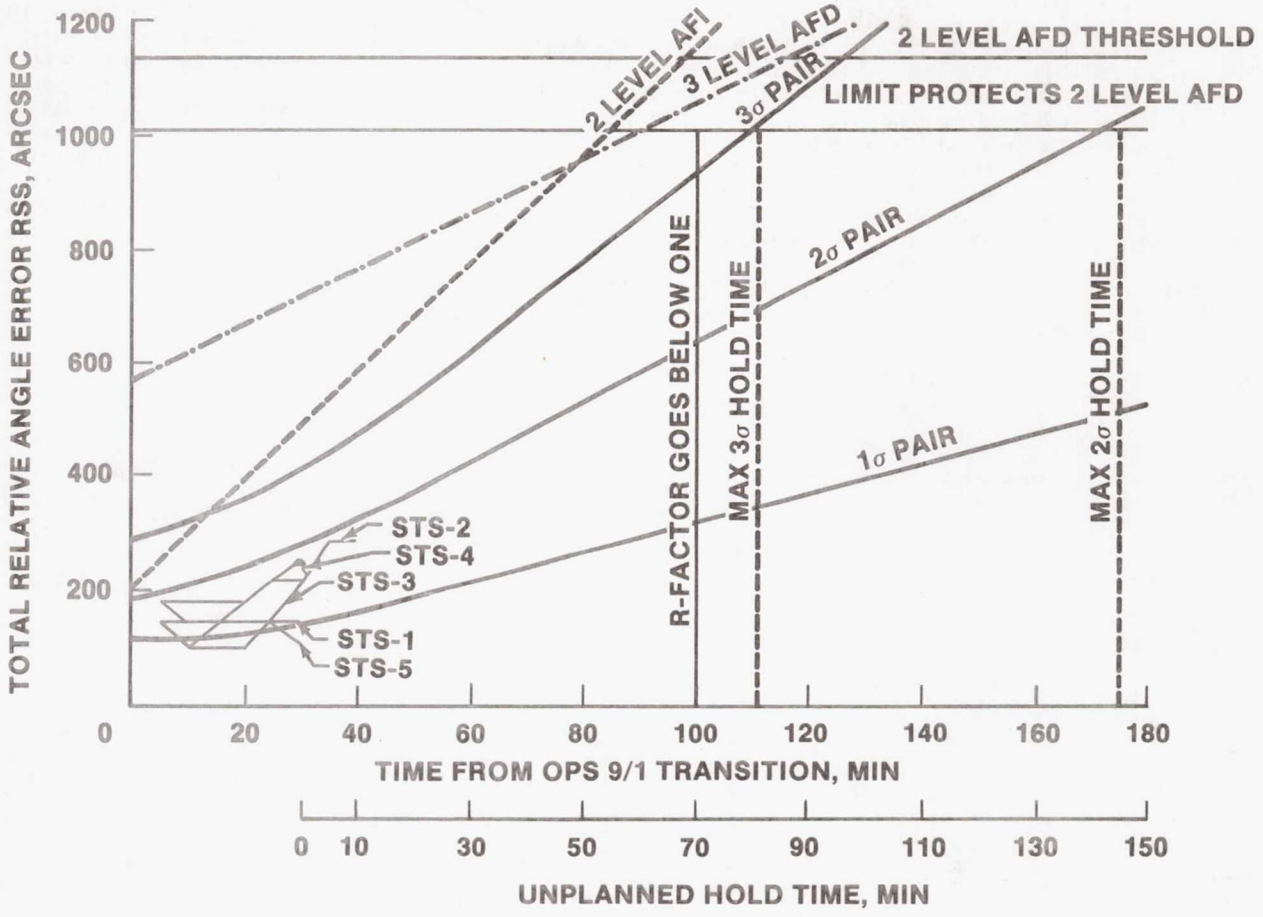

FIGURE 6.- PRELAUNCH IMU DRIFT MISALINEMENTS.

MEASUREMENT REDLINE

Protecting the threshold against false alarms implies the observed instrument error is below the threshold after making allowance for observation errors. The measurement error bound can be estimated analytically. The granularity of the display is 0.01 deg/axis and the measurement error can be no more than 0.01 deg/axis about three axes or 0.017^{o}. The Mission Control Center algorithms compute pairwise IMU errors using functions of gimbal angles the readout quantization of which is 20 arc-seconds per axis, and the expected measurement error would be no less than 20 arc-seconds about eight axes of two four-gimbal platforms or 0.0157^{o} (RSS of 20 taken eight times). An additional error of 65 arc-seconds is budgeted for the resolved errors and gimbal pivot misalinement which shows up during the post-lift-off tower-clear roll maneuver as observed in Monte Carlo simulations of the ascent flight profile.

An IMU performance redline is established as follows:

The threshold	0.316^{o}
Less measurement error	0.017^{o}
Less tower-clear roll error	0.018^{o}
Limit line	0.281^{o}

This limit line, plotted in figure 6, is violated by a 3σ IMU pair 111 minutes after platform release.

MCC MONITORING OF EXTENDED HOLDS

Because the RM threshold could possibly be violated at 80 minutes into a hold, the Flight Operations Directorate decided to monitor relative performance to protect the threshold knowing they could terminate an unplanned hold earlier; otherwise, the program would continue the present baseline which allows 90 minutes of unplanned hold.

VELOCITY UNDERLIMIT BITE

The loss of IMU velocity output is one area of software transient protection that is not adequately covered. The current understanding of the failure mode is that all three accelerometer counters freeze. The IMU SOP will then output the accelerometer bias calibration terms only. Therefore, wrong isolations or dilemma can occur.

The VUL BITE was designed to detect the loss of velocity output. The VUL BITE is the only BITE that compares one IMU against another in order to make a decision. The VUL BITE uses the compensated delta velocities to perform the test so the frozen counters are masked by the compensation terms. This test works only in acceleration environments greater than 60 000μg (i.e., not during OPS 2) and the test is limited to IMU's having accelerometer bias calibration terms less than 50 000μg. Commfaulted IMU's are not excluded from this test. The crew select/deselect switch determines the test candidacy. If a loss of velocity output occurs when the VUL BITE is disabled, a dilemma or wrong isolation will occur because of the large accelerometer bias compensations.

A solution to this problem is to check the uncompensated velocity counts for each IMU. The revised VUL BITE test would then be performed at a rate such that the smallest accelerometer bias compensation term yields at least one count.

If the sum of the squares of the velocity counts for all three axes is less than three, the counters have frozen and that IMU has a VUL failure. This test assumes that the velocity counter bit toggling will cause at most one count per axis during the period of consideration. This proposed VUL BITE test is valid during all flight phases. The only problem with the revised BITE is that transient BITE false alarms can occur if the environment exactly matches the accelerometer bias compensation terms. The revised VUL BITE test was felt to be an expensive software change and the community was willing to accept the loss of failure coverage (ref. 6).

VELOCITY SELECTION FILTER RECTIFICATION TRANSIENTS

Another area of inadequate software transients protection is in the velocity selection filter. The velocity-data-good logic was designed to protect navigation from large constant IMU failures. However, during STS-3 on-orbit, a new IMU failure mode was observed (ref. 7). IMU 3 experienced a large transient bias shift (34 750μg) followed by a smaller constant accelerometer bias (500μg). If this failure had occurred at the two-IMU level, the selection filter would have correctly performed the temporary downmode to the one-IMU level for the duration of the transient. If the transient had occurred between rectification cycles, navigation would have experienced the full effect of the transient (ref. 8).

One solution to the problem of transients polluting navigation is to exclude temporarily deselected IMU's from the selection filter until a new set of rectification biases has been computed. Deselected IMU's refer to commfaulted IMU's or any temporarily downmoded IMU due to a velocity-data-good violation. This procedure will prevent the transient from polluting the selected velocity and thus the single-string navigation. Currently, a crew-reselected IMU is treated in this manner. However, this solution does not protect the entry three-string navigation since it does not use the selected velocity. The ground support console operators will have to monitor the three navigated states to protect navigation from transients.

This solution has been approved for software implementation as CR 59308A on release 2X/XX.

FLIGHT TIME-LINE IMU RM EFFECTS

The values of the IMU RM fault detection and isolation thresholds are a function of certain mission events. The attitude thresholds ramp from either OPS 9/1 transition (platform release) or on-orbit alinements. The velocity thresholds are constant before lift-off, jump to a higher value during ascent, and drop to a low value until MM 304 (entry interface minus 5 minutes). At MM 304, the velocity thresholds begin to ramp until a high constant value is reached. The threshold values were determined by instrument performance, guidance constraints, and IMU platform skewing geometry.

The velocity thresholds are insensitive to mission time-line changes since the triggering events always occur at roughly the same times. However, time-line changes are critical to the attitude thresholds since they are reset at each on-orbit alinement. The attitude fault detection threshold ramps to a constant value while the fault isolation threshold continues to ramp. Consequently, after about 5 hours, all two-IMU level attitude failure coverage is lost (fig. 7). The attitude thresholds were set based on the last onboard alinement occurring approximately 70 minutes before the deorbit burn. The attitude RM covered about 60 percent of all failures not covered by BITE. (BITE covers about 90 percent.)

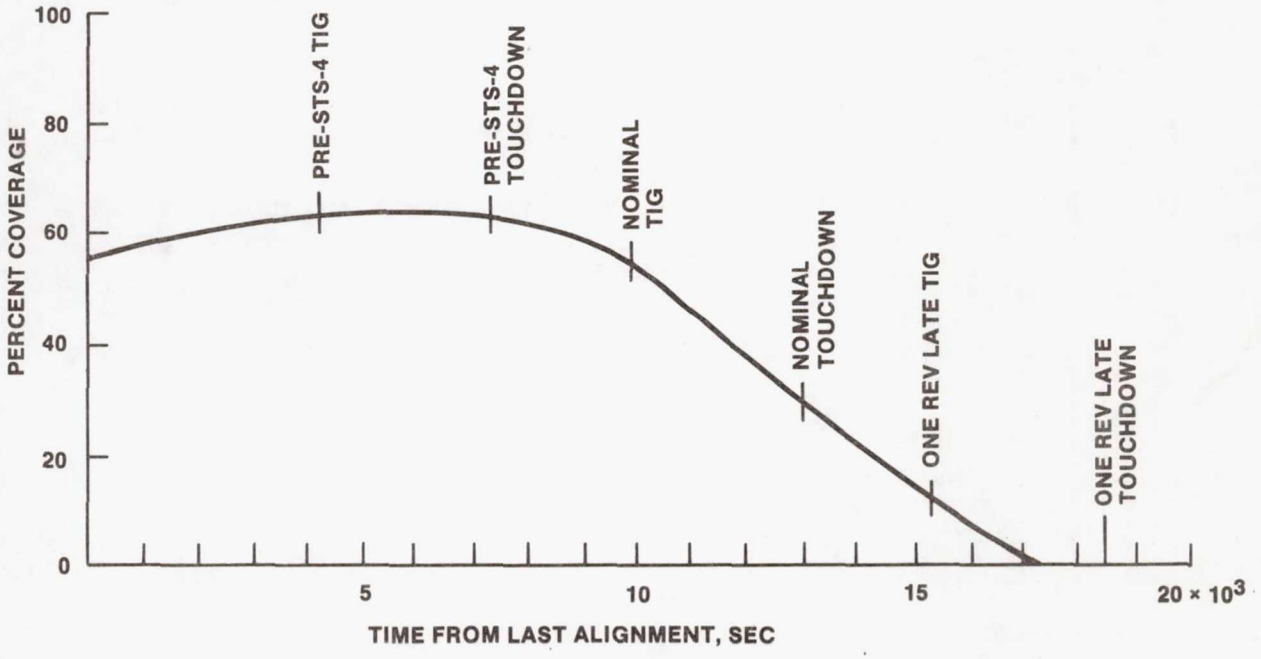

FIGURE 7.- TWO-IMU LEVEL ATTITUDE FAILURE COVERAGE.

Before STS-4, it was decided to perform the last onboard alinement one revolution earlier than on previous flights (165 minutes before the deorbit burn) to ease the crew workload. Unfortunately, this decision caused a degradation of the two-IMU level attitude failure coverage. (See fig. 1.) The Entry Flight Techniques Panel decided to leave the time line alone if there are three good IMU's. If there are only two good IMU's in the system, the last onboard alinement will be performed 70 minutes before the deorbit burn, as it was done previously (ref. 9).

The other flight time-line attitude threshold concern is for a one-revolution-late deorbit. All two-IMU level attitude failure coverage will be lost before touchdown (ref. 10). In this case, an IMU-to-IMU alinement should be performed about 70 minutes before the deorbit burn.

The entry attitude threshold criteria preclude changing the values without degrading the RM coverage even further. Since the IMU's have been behaving well on orbit and during entry for the first six flights, it may be possible to reevaluate the 1σ drifts and lower the thresholds, thus regaining RM attitude coverage.

REFERENCES

1. Space Shuttle Flight and Ground Systems Specification. JSC-07700, Vol. X, Rev. C, Para. 3.3.1.2.3.1.2.

2. Thibodeau, J. R.; and Bauer, Steven R.: Redundancy Management of the Space Shuttle Inertial Measurement Units. AIAA Guidance and Control Conference, Aug. 9, 1982.

3. The High Cost of Worrying Improbable Possibilities. JSC Memorandum LA15(81-51B), Sept. 28, 1981.

4. Redundancy Management Flight Systems Software Requirements. STS 83-0010, NASA Lyndon B. Johnson Space Center, 1983.

5. Bennett, D. E.: QMVS Evaluation at SES. Rockwell International, Internal Letter No. 283-210-82-019, June 14, 1982.

6. Schneider, H. E.: Velocity-Under-Limit BITE Proposed Fix. MDTSCO Transmittal Memorandum 1.4-TM-D2630-380, Feb. 12, 1982.

7. Gibbs, R. G.: IMU 3 Accelerometer Error During STS-3 On-Orbit. MDTSCO Transmittal Memorandum 1.4-TM-D1730-72, June 22, 1982.

8. Schneider, H. E.: Velocity Data Good Logic. MDTSCO Transmittal Memorandum 1.4-TM-D1730-660, July 21, 1982.

9. Schneider, H. E.: IMU-RM Impact of Early Entry Alignment. MDTSCO Transmittal Memorandum 1.4-D1730-681, Aug. 6, 1982.

10. Schneider, H. E.: IMU-RM Impact of One Rev Late Deorbit. MDTSCO Transmittal Memorandum 1.2-TM-FM53A01-457, Mar. 29, 1983.