Decentralized Filtering and Redundancy Management for Multisensor Navigation

THOMAS KERR, Senior Member, IEEE
Intermetrics Inc.

Failure detection and redundancy management is discussed for avionics applications of integrated navigation involving coordinated use of multiple simultaneous sensor subsystems such as GPS, JTIDS, TACAN, VOR/DME, ILS, an inertial navigation system (INS), and possibly even Doppler AHRS. A brief high level survey is provided to assess the status of those techniques and methodologies advertized as already available for handling the challenging real-time failure detection, redundancy management, and Kalman filtering aspects of these systems with differing availabilities, differing reliabilities, differing accuracies, and differing information content/sampling rates.

Following the status review, a new failure detection/redundancy management approach is developed based on voter/monitoring at both the raw data and at the filtered-data level, as well as using additional inputs from hardware built-in-testing (BIT) and from specialized tests for subsequent failure isolation in the case of ambiguous indications. The technique developed involves use of Gaussian confidence regions to reasonably account for the inherent differences in accuracy between the various sensor subsystems. On-line estimates of covariances from the Kalman filter are to be used for this purpose (when available). A technique is provided for quantitatively evaluating both the probability of detecting failed component subsystems and the probability of false alarm to be incurred, which is then to be traded off as the basis for rational selection of the thresholds used in the automated decision process. Moreover, the redundancy management procedure is demonstrated to be amenable to pilot or navigation operator prompting and override, if necessary.

A structure to accommodate differing rates of subsystem assessment and tally is developed and alternative designs for navigation architectures are offered based on likely subsystem utilization and newly emerging concepts in decentralized Kalman filtering. Many of the decentralized filtering concepts are only now economically feasible for real-time implementation due to recent availability of commercial parallel processing chips and/or VHSIC compatible systolic array versions of all the requisite algorithms and transformations necessary to support such Kalman filter mechanizations in a few chips.

Manuscript received April 7, 1986; revised July 22, 1986.

An earlier version of this paper appeared in *Proceedings of the National Technical Meeting of the Institute of Navigation (ION)*, San Diego, Calif., January 15–17, 1985.

Author's present address: MIT Lincoln Laboratories, 244 Wood St., Lexington, MA 02173.

0018-9251/87/0100-0083 $1.00 © 1987 IEEE

I. INTRODUCTION

In recent years, several studies and projects (e.g., [52, 55–57, 60, 83, 85, 99, 103, 109–111, 115, 119, 122–125, 135–141, 160–167, 182, 186, 189, 202, 231, 238, 239]) have sought to reasonably automate the on-line handling of avionics navigation system information reliance involving several sensor subsystems. Such navigation configurations usually include an inertial navigation system (INS) gyros and accelerometers, in conjunction with use of simultaneous alternative navaid sensor subsystems such as GPS, JTIDS RelNav, TACAN, VOR/DME, ILS, radar altimetry, and possibly even Doppler AHRS.

In order to exhibit fast reaction to changing conditions and to avoid overwhelming the navigator with useless intermediate check data, the multisensor navigation should proceed automatically while conceding to navigator or pilot override, when desired [82]. Such a system is generically denoted here to be a semiautonomous multisensor navigation (SMN) system.

Development of fault tolerant computers [67] and software architectures [119] have been proceeding at a rapid clip and many standard fault detection/avoidance techniques are beyond reproach such as those common techniques that were developed for detecting and handling failures in hardware/design (e.g., power supply monitoring, parity bit checking, use of write and overflow protection, use of error correcting coding, interface signals, watch-dog timers, redundant CPU processors, etc.), for detecting failures in software (e.g., handling of worst case conditions or loading in test program diagnostics, memory check-sum tests, walking "1" test, etc.), and for detecting failures in hybrid imbedded applications involving both hardware components and software/computers (e.g., end-to-end and wrap-around testing techniques). Although standardization of design and production standards is proceeding [111] for the abovementioned failure detection techniques, it is somewhat disconcerting that apparently no unanimity of opinion exists for the utility of alternative failure detection techniques that have evolved for detecting failures in systems whose operation is described by dynamic equations (e.g., differential or difference equations with possible additive stochastic or random noise terms present) which typify navigation system models [106]. A brief overview survey of the status of these navigation-related failure detection techniques is offered in Section II as a preferred approach, to the SMN application is sought.

Although a failure modes and effects analysis (FMEA) is crucial to any endeavor of this type ([111, 176, 54]) in order to match the techniques used to the expected types of failure modes and signatures, this topic is not included in this general discussion because its utility depends entirely on what specific assortment of sensors are to be included. Preliminary assessments of the FMEA for typical SMNs are offered in [66, 110, 52], while a detailed assessment is available in accessible company studies [115].

Based on the failure detection status assessment, a few of the more favorably represented techniques are combined together in Section III to result in a noise-robust voter/monitor that is offered as a preferred approach for handling the general SMN application and which is shown to also comply with system and computer architectural, timing, and resource constraints typically encountered. Section IV describes the lure of recently evolving decentralized Kalman filter mechanizations and demonstrates how the Speyer formulation [86] (along with its recent refinements) can be adjoined with the voter/monitor approach of Section III for failure detection isolation and reconfiguration (FDIR), developed here as a recommended approach for handling the SMN application. Critical navigation related issues associated with its performance and implementation are also addressed. The aspect that makes such sophisticated software and algorithm implementations currently economically feasible is their compatibility with systolic array implementation associated with VHSIC, as discussed in Section V. A brief summary and associated conclusions are offered in Section VI.

II. SURVEY OF CURRENT FDIR RESULTS FOR NAVIGATION

Failure detection and failure isolation are common problems in complex navigation systems. In general, failure detection requires continuous vigilant monitoring of the observable output variables of the system. Under normal conditions, the output variables follow certain known patterns of evolution within certain limits of uncertainty introduced by slight random system disturbances and measurement noise in the sensors. When failures occur, the observable output variables deviate from their nominal state-space trajectories or evolutionary pattern. Most failure detection techniques are based on spotting these deviations from the usual in the observable output variables.

In modern avionics systems, failure detection and built-in-testing (BIT) are the cornerstones used to signal when the principal path of information reliance or data flow should be switched to an alternative backup path to preserve proper overall system performance. While on-line BIT is usually used to rapidly uncover catastrophic or hard-over failures, other more sophisticated techniques (based in modern control/estimation/decision theory) are utilized to detect more subtle or "soft" drifting-type failures that do not necessarily cause the system to shut down entirely but may still considerably degrade system performance with the passing of time unless remedied.

The following considerations are some of the factors that contribute to an overall failure detection/failure isolation/system reconfiguration policy:

- nature of the soft failure (i.e., its type and severity);
- observability of the failure's effect within the measurement (i.e., degree of perceptibility);
- length of time required to accumulate enough data to register the presence of the failure in the presence of background disturbances such as quantization effects, sensor noises, maneuvers, and even enemy jamming;
- degree of distinguishability from other types of failures for unambiguous failure isolation (cf., [224]);
- ease of corrective actions (e.g., switching to an alternative analytically or functionally redundant path on-line or postponing repair until back at the maintenance depot).

However, the action of failure detection is fundamental to every system reconfiguration policy and the technological area of failure/event detection is undergoing rapid change as new ideas enter the field. These emerging approaches are discussed next.

The reference bibliography [1–80, 159–167, 174, 175, 190–196, 201, 202, 204–206, 214–216, 224, 237, 238] should serve as substantiation of the extent of our coverage of significant failure detection events that have occurred since the appearance of the first paper in this area 19 years ago [1]. More explicit comments on the advantages, disadvantages, and contribution of most of these approaches is provided in an accessible report [115]. Since the perspective taken here is relevance of the automated failure detection approach to navigation systems, the results and conclusions of this survey are germane only to the navigation aspect of SMN.

Rather than dwell on the details of each study, the commendable milestone accomplishments, or originality of design, the limited objective here is merely to summarize the advantages and disadvantages as they relate to SMN and to evaluate these techniques with regard to effectiveness of performance, ease of implementation, risk incurred when required parameters are apparently not yet available, lack of reasonable experimental confirmation in flight tests of early theoretical predictions or perhaps promising results but from oversimplified low-order simulations, etc. By observing *all* of the significant advantages and disadvantages of the various existing techniques and by carefully avoiding the pitfalls, it was hoped that the proper approach for a FDIR design for SMN navigation could be extracted from existing tools to avoid unnecessarily high development risk.

The somewhat hardnosed survey of [48] reveals the following drawbacks of several current approaches to failure detection:

- applicable only to time-invariant systems, while navigation systems usually have time-varying linear error models [181];
- demonstrated only for low-order systems (1, 2, or 3 states), while actual navigation system linear error models are considerably more complex;
- demonstrated only when system model is *identical* to Kalman filter model, while practical implementations use reduced-order filters with imperfect nonwhite residuals even in the unfailed nominal case;

- demonstrated only for extremely large signal-to-noise ratios (SNRs) rather than for SNRs that would realistically be encountered;
- frequently, indications are missing entirely of what underlying SNR actually is in many research studies (however, a recommended SNR evaluation procedure for applications of this type is offered in [17 and 48, pp. 971–973];
- claims of *perfect* detection with no false alarms (clearly a physical impossibility since all detection decisions involve obtaining a balanced acceptable tradeoff between undesirable false alarms and undesirable missed detections);
- only simple failures (such as biases directly in the measurements) are considered exclusively in many studies rather than considering the occurrence of failures more challenging to detect yet more frequently occurring in practice;
- no theoretically rigorous methodology available for determining decision threshold specification, as needed for a complete comparison of test statistic against decision threshold in making failure/no-failure decisions;
- no rigorous quantification of underlying probabilities of false alarms and miss;
- no way to account (and therefore compensate for) serial (time) correlations in the estimates as they affect the test statistic (as functions of these filter estimates) except as developed in [18].

These considerations also served as very specific criteria for initial weeding and rejection of particular approaches from further consideration for the SMN application as likely to be too high a risk.

One particularly important criterion from the above list that is frequently overlooked but is especially appropriate as a test against reality for failure detection approaches proposed for avionics navigation applications is further emphasized here now. In general, there is an underlying truth model of fairly high dimension that completely describes the detailed error evolution of the augmented navigation system consisting of several components. However, typically only a reduced-order Kalman filter (using just the most significant states) is implemented on-line for real-time applications due to the practical constraints on allowable computational delay and computer memory available. Many common failure detection approaches were derived using system descriptions and Kalman filters that are of the same dimension as the truth model or full navigation error model and the rigor of the derivations underlying the detection method critically depend on the filter residuals being white and unbiased in the unfailed nominal situation that is typically assumed to be the prevalent mode of operation. In practice, however, the filter residuals can be nonwhite or biased for the following reasons:

(1) because a failure occurred;
(2) because a bad measurement was received (i.e., presence of statistical outliers or data gaps);
(3) because of the standard use of a reduced-order suboptimal filter in the application ([81, 82]) as is routinely implemented in navigation applications due to constraints on computational capacity available.

Any failure detection approaches that do not explicitly acknowledge the last two reasons above as possibilities consequently incorrectly attribute any nonwhiteness encountered in the application to be solely due to the occurrence of a failure. Just raising the decision threshold in order to compensate does reduce false alarms but makes the test less sensitive to actual failures and can cause missed detections when failures do occur.

Other more robust approaches to failure detection have been developed which do not unrealistically depend on assumptions of whiteness/unbiasedness of the associated filter residuals or even on the Gaussianess of the system and measurement noises. Three such robust yet failure sensitive approaches to redundancy management are

(1) voting between three (or more) comparable components;
(2) midvalue selection (between three comparable components);
(3) reliance on parity equation checks between either identically redundant systems or functionally redundant sytems or combinations of systems which together cover the function of another system (known as analytically redundant systems).

Recent generalizations of each of the above techniques to dynamical system models and navigation applications exist as, respectively, [78, 50, and 51] (as further explained in [38, 49, and 159]). Voting is further discussed in what follows.

Based on over a decade of industrial experience in developing an automated real-time failure detection technique for navigation systems (e.g., [14–22]) and in monitoring the status of emerging techniques from a practical viewpoint (as evidenced in [48], as has been further augmented for SMN here), an overview is offered in Table I on pertinent aspects of the following four failure detection approaches:

(1) generalized likelihood ratio (GLR) tests;
(2) sequential probability likelihood ratio tests (SPRT);
(3) generalized likelihood test (GLT)/maximum likelihood detector (MLD),
(4) voting tests.

Both the advantages and disadvantages of each of the above approaches are discussed in Table I in somewhat general terms; then a recommendation is provided on how to conservatively proceed in the SMN application in order to reduce the development risk.

In lieu of the preponderance and nature of disadvantages over advantages listed in Table I, it is

TABLE I
Overview Considerations of Four Popular Techniques for Detecting Soft and Medium Failures in Dynamical Systems

APPROACH	ADVANTAGES	DISADVANTAGES
Generalized Likelihood Ratio (GLR)	• Appealing analytic framework appears to suite the requirements of the detection task; • Widely embraced and understood by many investigative researchers; • Ostensibly applicable to aided navigation systems already utilizing a Kalman filter [10], [12].	• Claims of being an optimal test not substantiated [48]; • No viable method yet provided for specifying decision threshold needed for comparisons against on-line GLR test statistics in making failure/no-failure decisions [48]; • Deemphasis of (necessary) decision thresholds in recent application studies [23], [24], [43]; • Untenable computational burden as exact GLR implementation needs an increasing bank-of-Kalman-filters; • Computationally tractable variations as approximate "practical" GLR implementations incur intolerable degradations [48].
Sequential Probablity Likelihood Ratio Test (SPRT)	• Can be used in conjunction with a Kalman filter; • Approximate decision thresholds are easy to calculate and relate to underlying probabilities of error (but only for an underlying simple binary hypothesis model); • Well-known philosophy of operation stemming from 1947 roots in Wald's SPRT [2].	• Model mismatch: Failure detection is more involved than just a simple binary test (although treated with Wald's SPRT as such). Unknown time and magnitude of failure dictate that a mixed hypothesis is more appropriate. • Resetting to midpoint between failed and unfailed decision thresholds (as in [6], [74]) is an ad hoc mechanism. • Use of SPRT technique requires prior knowledge of failure, magnitude (as being either large or small) - an unrealistic constraint/assumption in the SMN application. • Use of prior decision tests (in one implementation [7]) as triggers to indicate when SPRT processing is to commence is ad hoc and introduces additional unknowns as cross-correlated effects of back-to-back tests and appropriate decision thresholds being not yet quantified. • Well-known decision threshold evaluation methodology and probability of error quantification for Wald's 1947 SPRT for binary hypothesis unrelated to actual mixed hypothesis situation of failure detection, as now realised but rigorous threshold determination is currently intractable [62]. • Recent 1982 failure detection investigations [62] do better model matching in using Shiryayev's SPRT instead of Wald's version, but associated computational burden of Shiryayev's updated version is equivalent to "curse-of-dimensionality" in exact implementation of Dynamic Programming/Viterbi algorithms without trellis pruning and should therefore be avoided. • 1980 NASA flight tests of SPRT [60] apparently inconclusive since (supposed constant) level of failure decision thresholds remainded in flux and were even modified following the simulated missions (as indicated in [60]).
Generalized Likelihood Test/ Maximum Likelihood Detector (GLT/MLD)	• Reasonable computational burden; • Uses theoretical framework consisting of maximum likelihood estimation/identication techniques; • Can be reformulated to check parity vectors [49], [p. 21, 65] to assist in failure isolation; • 1983 modification [51] drastically reduced otherwise substantial computer burden encountered in calculation of adaptive decision thresholds responsive to maneuvers. Such use reduces the number of anomalous excursions of test statistic above an uncompensated decision threshold, thus reducing the number of false indications of system/component failure.	• Full adaptation of decision thresholds to accomodate all anticipated naturally occurring transients to be encountered during maneuvers eventually became larger computational burden than original objective of providing on-line calculation of test statistic. Time-varying compensation procedure [51] not fully tested out and a separate (second) Kalman filter was called for in implementation. • Flight tests [40] indicate that INS failures take 10-15 minutes to detect and isolate while preliminary investigations [10], [11] predicted detection in 0.5 minutes, detection in a few seconds [52], and instantaneous detection [12].
Voting	• Can be applied either directly to the raw measurements prior to possible contamination from subsequent processing or applied to subsequently filtered and therefore further refined estimates of the sources of potential problems; or applied to both. • Voting tests can be posed in a form that is compatible for representation as parity vector/table cross checking to simplify failure isolation (see pp. 20-21 of [65], [76]). • To account for differing accuracies of contributing components, parity equations can be modified from merely being equated to zero (as expected for ideal exact agreement between sensors) to being equated to a quantity that is operationally equivelent to zero (for all practical purposes) by using variable decision thresholds for comparison (which can therefore provide sufficient additional leeway by accounting for expected standard deviations of each participant along with components to account for noise and maneuvers). • Sophisticated generalization of voting test (as [78]) operates on the output of the Kalman filter and gently deweights a possible dissenter's contribution from overall navigation solution based on instantaneous degree-of-departure without breaking system structure or priority of data flow.	• Requires additional logic for implementation (but as compared to the other approaches mentioned above, the associated computational burden is minor). • Voting on raw measurements is less refined, may lack the necessary precision for unambiguous isolation, and may take too long to detect and isolate failures. • Voting on filtered data may incorporate and smooth over deleterious effects of faulty processing and therefore contaminate conclusions.

prudently recommended that the use of GLR be avoided in the SMN application even though new (but as yet untested results [142]) are on the horizon. Additional reservations are enumerated in [211]. For similar reasons, it is recommended that the use of SPRT be avoided in the SMN application.

Due to the nature of the disadvantages cited in Table I, it is recommended that aspects of GLT/MLD testing continue to be explored as a basis for reasonableness tests or as tests on suboptimal but nominal residuals. It is highly desirable to avoid use of a second Kalman filter for threshold adaptation if possible; otherwise it is an unreasonably large computer burden for the SMN application. Finally, due to the preponderance and nature of the advantages over the disadvantages listed in Table I, it is recommended that voting tests be used in the SMN application. In an attempt to temper the last two disadvantages cited for voting in Table I, it is recommended that

- voting be used *both* at the level of raw measurements (where computationally expedient maximum likelihood estimators can be utilized that take into account the effect of measurement uncertainties [114]) and that voting tests also be used on the filtered estimates;
- resulting voting decisions be further corroborated between the two levels when applicable (as demonstrated in a methodology offered in Section IIIA).

Ample real application precedents for the use of a voter-monitor structure in flight control and navigation projects exist [53, 55]. Kubbat [53] indicates how modern failure detection and isolation techniques can ameliorate the problem of handling failures in avionics applications for guidance and control objectives involving use of MIL STD 1553 data bus with redundant control and sensor units.

On the plus side, Kubbat [53] uses a voter-monitor type structure for three units whenever possible and forces triple redundancy. Kubbat [53] worries about multiple failures and how to handle them in a failsafe fashion and is able to point out ambiguity when only two systems of comparable accuracy are present. He recognizes situations where the algorithm can tell that something has failed when systems disagree, but not which one [53]. Unfortunately, Kubbat was not specific about how failures will be detected other than by majority rule voting. No consideration was offered in [53] of setting decision thresholds and effects of noises making even identical unfailed sensors disagree slightly (as *is* handled here in Section III).

Folkesson [55] describes various computer-based techniques for monitoring flight control system (FCS) components such as sensors, servos, and the FCS computer itself. The advantages of [55] are

- presents very efficient algorithmic implementation of midvalue select methodology;
- stresses use of voting and triplex sensors;
- stresses comparison monitoring of sensors;
- stresses use of reasonableness tests (but not necessarily filter-residual based).

A disadvantage of [55] is that it is not specific about how to detect soft failures or how to isolate whether one triplex sensor has already failed and another subsequently fails.

While not strictly a failure detection technique, but a failure protection/reconfiguration technique, the midvalue select algorithm has valuable qualities which make it desirable to use, as is discussed in [55]. The MVS property of never selecting the worst signal out of three is extremely valuable when one sensor fails. The failed sensor will be the extreme or worst one and therefore immediately detected and isolated. A recent generalization of the midvalue selection technique is offered in [159].

A recent survey [167] by Merril on the status of GLR, SPRT, MLD, and Bayesian hypothesis testing has a tone that conveys considerably more optimism than expressed here in the eventual performance of these same failure detection techniques, but the application arena is fundamentally different for his jet engine turbine control problem than for our SMN navigation sensing problem. Jet engine control apparently has a much lower dimensional (possibly time-invariant) self-contained state-variable truth model with several (4 to 10) control variables as deterministic inputs, thus allowing the system to be exercised or put through its paces to an extent as a type of so-designated "dual control" problem where some of the control energy can be expended during a preliminary learning period to enhance knowledge of underlying parameter values and state variables and whether it is indicative of nominal or failed performance. On the other hand, the aggregate SMN problem usually has a considerably higher dimensional time-varying underlying truth model with no cleanly defined deterministic inputs available for exercising control. Nor are all aspects of the truth model for the types of SMN systems under consideration totally self-contained within the aircraft since conditions relating to movement and location of any external GPS satellites or JTIDS RelNav net participants or VORTAC transmitters, etc., or conditions of the broadcast transmission medium (e.g., interference/jamming) have a considerable effect on the accuracy of the nominal SMN performance. These fundamental differences in the nature of the two applications notwithstanding, the survey of [167] displays what it designates as successful performance results but does not include examples that illustrate proper performance using any of the several decision thresholds that were claimed to have been developed within the programs. Indeed, in the conclusion of [167], it is identified that to date detection of soft failures has been accomplished in the jet engine control application by simulation only, and even that without resounding success. As indicated in [167, Table 4], unacceptable detection performance was indicated for almost 50 percent of the types of soft failures for which detection was sought. Merril [167] indicates that further gains in failure detection performance likely hinge on success in the pursuit of adequate reduced-order models. The need for adequate reduced-order filter models is also present in the SMN application and there is a need to heed several cautions [187], [242] that have been compiled in order to successfully obtain such models. The above background served as the starting point in developing a recommended FDIR approach for SMN applications as developed in Section III.

III. VOTER/MONITORING BASED ON GAUSSIAN CONFIDENCE REGIONS

Typical raw data information extraction goals of SMN-type military applications and resolution for common data representation in a geodetic coordinate system are depicted in the center column of Table II for each representative sensor of likely concern in navigation. The right column of Table II also depicts the information likely to be compared following filtering. A single table is used here since a common voter/monitoring methodology is developed in this section for handling both types of testing using just one algorithmic subroutine. The voter/monitor methodology advocated in this paper is depicted in conjunction with estimation in the overview of Fig. 1. The voter/monitor portion is represented simply as a block in this figure. The details are discussed below and further in Section IVB.

The first phase of the voter/monitoring methodology involves performing pairwise cross-comparisons of conformity of the navigation information provided by subsystem sensor i, to that provided by subsystem sensor j at the check time. The essence of this test is to compare the output realizations of the two designated sensors, which is of the form represented in the center column of Table II.

The actual outputs of the two sensors, denoted by $\hat{x}_i$ and $\hat{x}_j$, are encompassed by 1-sigma confidence regions (corresponding to the underlying Gaussian statistics) which also incorporate the associated mitigating uncertainty reflected by the covariances P_{ii} and P_{jj} (and

TABLE II
Information To Be Corroborated Pair-Wise in Voting Test

Component Subsystem	Available From Raw Data	Available from a Decentralized Filter Implementation[b]
PLRS	computed Position (3)	Pos(3)
JTIDS RelNav	Time-of-Arrival (TOA)[a]; Source Position (3)	Pos(3); Vel(3)
GPS	computed Position (3); Velocity (3)	Pos(3); Vel(3)
ILS	computed Pos (3)	
TACAN	computed horizontal Pos (2)	Pos(3); Vel(3)
INS	Position (3); Velocity (3);	Pos(3); Vel(3)
VOR/DME	computed Pos (3)	Pos(3); Vel(3)
Doppler	Velocity (3)	Pos(3); Vel(3)
Baro-altimeter	Altitude (as calibrated to reference)	used as damping[c]

Notes:
[a]JTIDS RelNav could be treated separately in the methodology of the raw data voting-test-of-compatibility as an exception because of the uniqueness of its information format (being dissimilar to all other formats).
[b]Linking lines indicate filter offered which combines more than one component subsystem.
[c]Since the Baro-altimeter is used to damp the INS, it is no longer available as an independent test for corroboration of altitude data except when the INS itself is barred from further participation in voting.

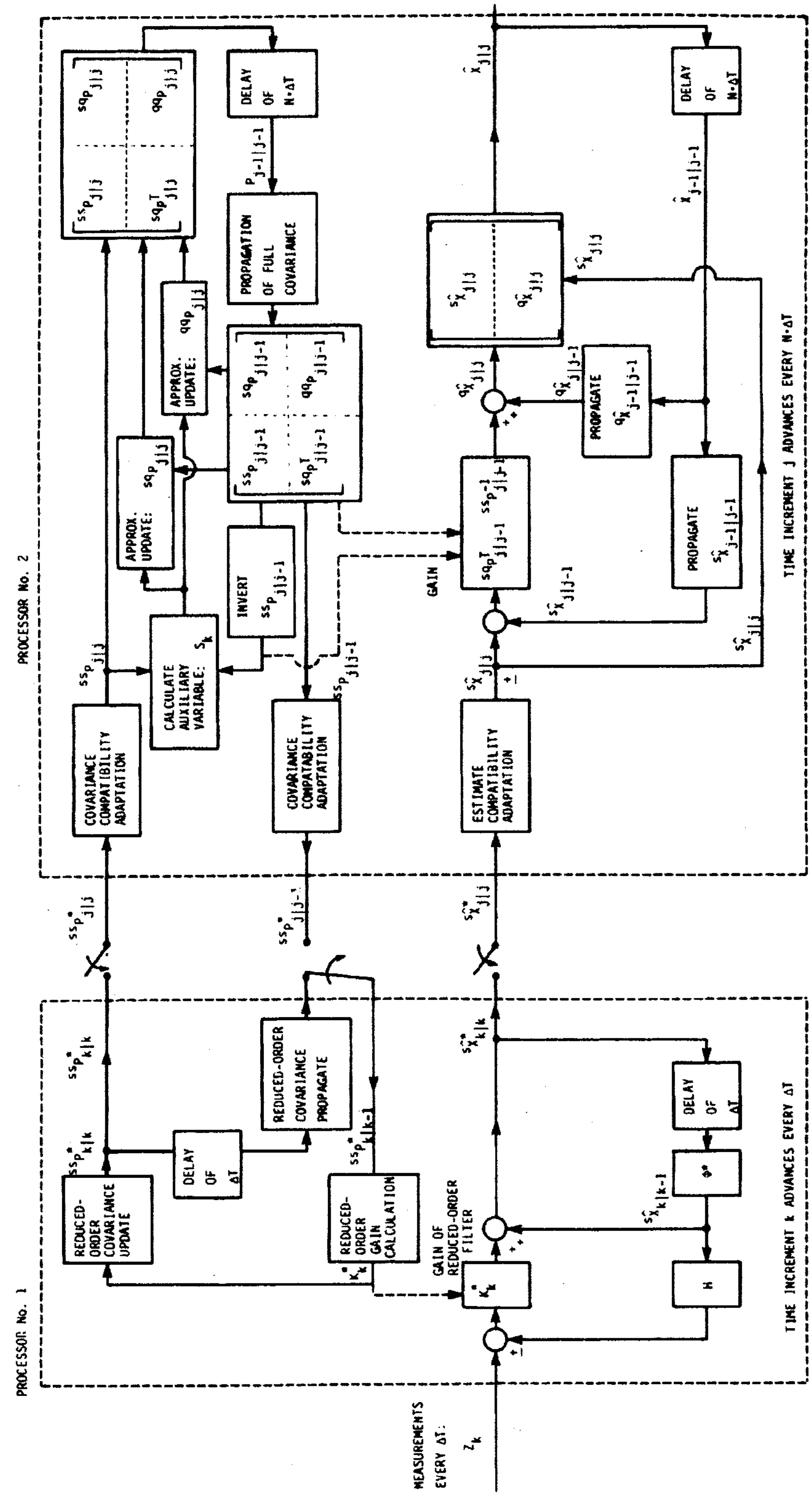


Fig. 1. Multirate two-filter approach: Reduced-order filter processes measurements at fast rate (ΔT) and hands over estimates to more accurately detailed full-order filter for processing at the slower rate ($N \cdot \Delta T$) [97].

P_{ij}, if available). The indication of true state is reflected from the dual perspective of considering the almost simultaneous measurements from these two different sensors. The two perspectives are considered to be consistent if there is, loosely speaking, sufficient overlap in the respective confidence regions. The explicit ellipsoid overlap test specified in [14]–[22] requires that the two specifically related inner product covariance matrices be of such a structure that their differences are positive semidefinite [213]. (While this condition was met in the application of [14–22], it is not met by general arbitrarily related covariance matrices such as are encountered in this SMN application. Consequently, the use of the technique of [14–22] is *not* recommended for this SMN application.)

The fundamental goal at each check time is to test the hypothesis on the consistency of the mean of two Gaussian distributions corresponding to the outputs of the two sensors under test. Ideally, the two means should be identical for both sensors and a standard assumption of the navigation errors being Gaussianly distributed is invoked. What is actually available for performing the test are the specific realizations at times "close enough" (as indicated by proximity of respective data time-tags) and the corresponding covariances as obtained from on-line evaluations to reflect the asserted accuracy of the two sensors. In line with common statistical practice, the test of consistency associated with the situation depicted in Fig. 2 involves calculating the following scalar test statistics at check time t_k:

$$l_{ij} = (\hat{x}_i - \hat{x}_j)^T [P_{ii} + P_{jj} - P_{ij} - P_{ji}^T]^{-1} (\hat{x}_i - \hat{x}_j). \qquad (1)$$

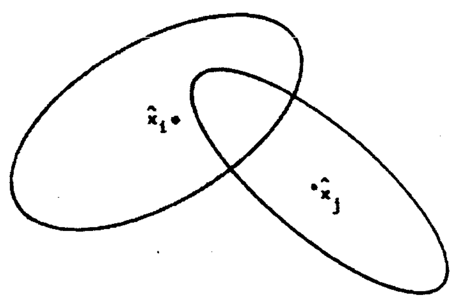


Fig. 2. Separate α-level Gaussian confidence regions about both realizations measured from sensors i and j.

Here, the data required to compute l_{ij} for any given sensor pair i,j are defined as follows:

- $\hat{x}_i$ — geodetic position or velocity components derived from sensor i raw measurement data (as in the middle column of Table II);
- P_{ii} — covariance matrix of errors in geodetic estimate of x_i;
- P_{ij} — cross-covariance of coupled errors in estimates from sensor i and sensor j;
- $TYPE_i$ — designator indicating type and dimension of data from sensor i available for comparison;
- $TIME_i$ — timetag associated with sensor i data.

For fault detection purposes, the test statistic comprises a random variable that has a chi-square distribution (see the appendix). Thus a constant decision threshold can be extracted from a chi-square table, representing the α-level of confidence (or α probability) that l_{ij} will be below that threshold value under normal (unfailed) circumstances. The α-level threshold value varies only with the number of degrees of freedom, or the dimension n of x_i and x_j. The proposed FDI test, then, is

$$l_{ij} \begin{cases} \leq L\text{threshold} \rightarrow \text{means are consistent} \\ > L\text{threshold} \rightarrow \text{fault indication.} \end{cases} \qquad (2)$$

An algorithm for performing this pairwise test of consistency for N sensor subsystems is offered in the flowcharts of Figs. 3 and 4.

Note that generally P_{ij} is zero for independent sensor outputs. However, the INS represents a significant exception to this rule when it is receiving feedback corrections from the navigation filter (the usual situation). In this case, the INS outputs are highly correlated with those of the most accurate radio navigation sensor (generally GPS) processed by the filter. At least three approaches are possible to handle this case. First, the INS can simply be barred from participating in the comparison tests when it is receiving feedback corrections. Second, the feedback corrections can be accumulated and propagated over time, so that they can be "backed out" of the INS solution at comparison points, yielding an independent data set. Third, the INS acceleration outputs can be integrated independently into velocity and position (outside the INS), such that an independent (noncorrected) solution results. In the second and third approaches, P_{ii} for the uncorrected INS must also be computed (state transition and process noise submatrices from the filter could be used).

The second phase of the voting methodology, illustrated in the flowchart between points B and E in Fig. 4, occurs where the votes are tallied on subsystem-to-subsystem disagreements. The subsystem that is implicated by the largest number of derogatory votes by the other subsystems as being out of kilter is flagged first, and the intensity of the negative voting (i.e., the total tally) is reflected in the designated entry on the SCORESHEET for that sensor (as further explained below). Once the votes are tallied and the prime culprit or subset of suspects are identified, the tallying process is repeated for the remaining participants, such that any secondary faults are similarly flagged. Note that it is quite possible for two sensors to receive an equal number of derogatory votes. In particular, two accurate sensors that disagree will vote against each other. If other sensors are either unavailable, or significantly less accurate, they will not be able to distinguish the real culprit and break the tie so both tied sensors will be declared suspect. Methods for further resolving such ambiguities are offered here based on follow-up checking of filter estimates for biases or trends of characteristic signatures corresponding to specific known failure modes, but at the expense of taking more time, but still not jeopardizing the mission by avoiding any undue reliance on suspect sensors.

An example of an auxiliary isolation test is useful here. Suppose that the GPS-derived position disagrees significantly with the (nonfeedback-corrected) INS position and that no other sensors are available, e.g., TACAN stations are too far away or are not accurate enough to identify the culprit. The GPS-INS position discrepancy vector Δp can then be projected along the four GPS satellite line-of-sight vectors. If the error continues to lie principally along one such vector as the

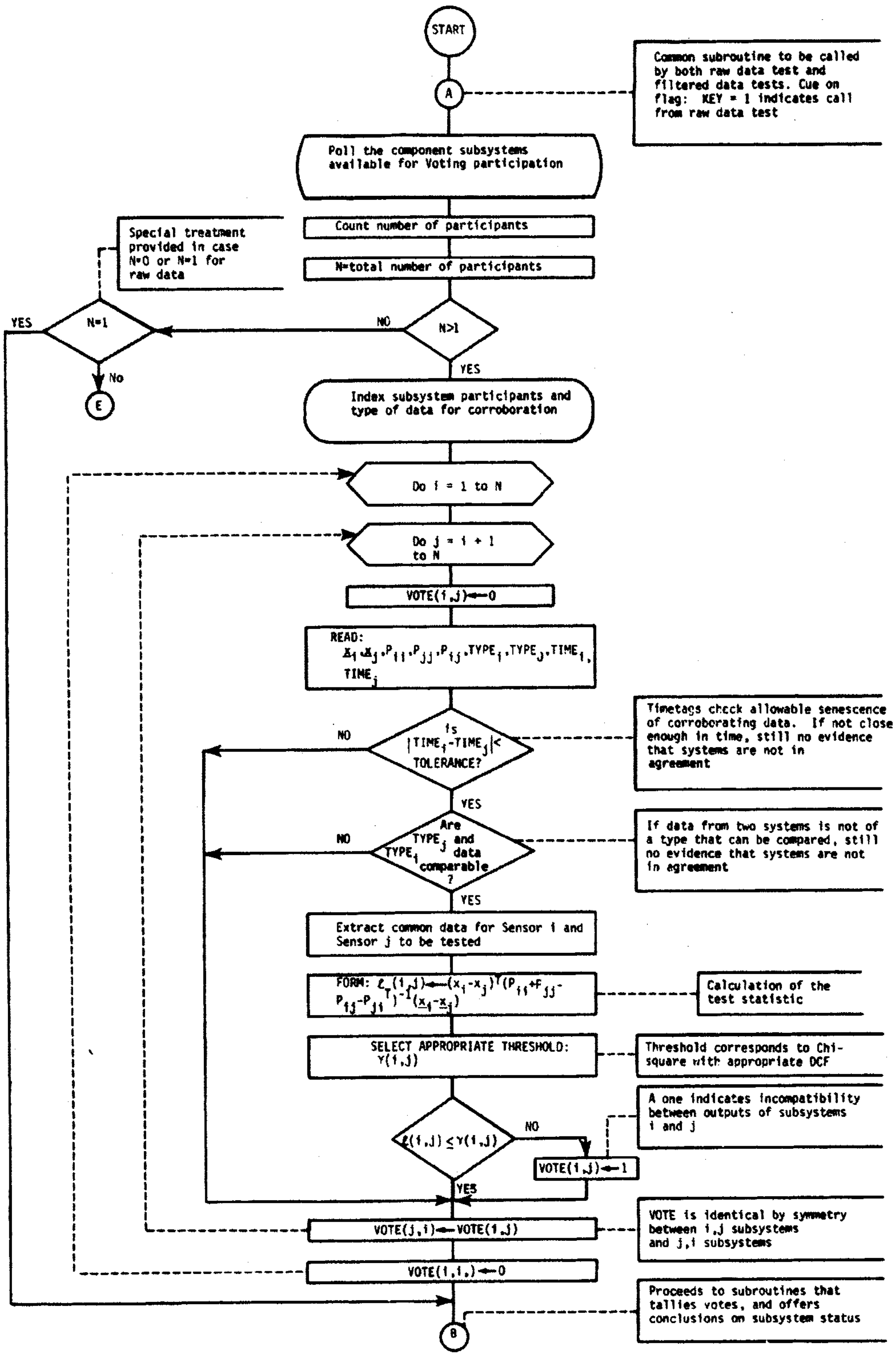


Fig. 3. Voter monitoring test, based on confidence regions, quantitatively accounts for uncertainty reflected in covariances due to environmental conditions.

user moves and the satellite constellation varies, and very little lies along the others, then that satellite (or tracking loop which is probably already sufficiently monitored by hardware bit for adequate distinguishability) is the likely error source.

A. On-Line Sensor Status Assessment

A computer memory datablock for each component sensor subsystem entitled SCORESHEET is to be asynchronously maintained. It is suggested that the

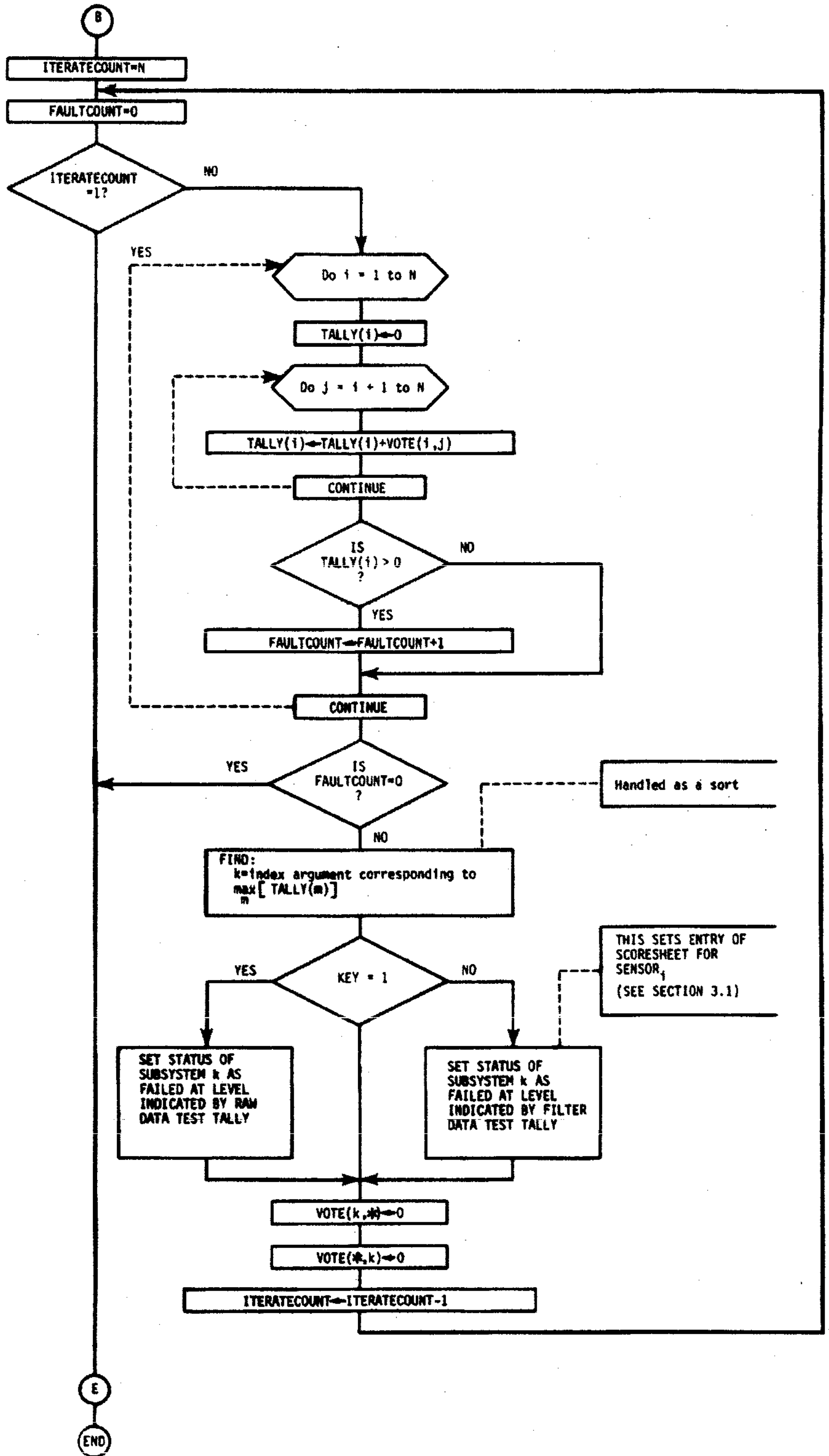


Fig. 4. Tallying the votes and tagging failed subsystems in order of worst ranked first, then retallying to find the next worst, etc.

SCORESHEET data block be composed of five data fields as indicated in Fig. 5 with a range of values possible to be attained, as described below the data block of Fig. 5. Following completion of the raw data test, the a_i field is set. Similarly for the b_i field, the c_i field, and the d_i field at whatever completion times are normally encountered for these functions and for each sensor. The importance of these four data fields as a convenient summarizing tool that enables fast response in detecting changes is illustrated better when BIT is considered in more detail as a representative example of how these data fields are effectively and efficiently used.

According to a prescribed rate schedule (of about every 1/2 second), the sensor status controller should check the SCORESHEET for every sensor participating in SMN and sum the fields to obtain SUMCHECK. Upon evaluating SUMCHECK, the sensor status controller can assess the sensor's status through a few quick logical calculations that proceed as follows:

If $\text{SUMCHECK}_i = 0$, then status of sensor_i is declared unfailed.

If $\text{SUMCHECK}_i \leq \text{FAILDECIDE}_i$ but higher than a lower tolerance threshold, then status of sensor_i is declared suspect.

If $\text{SUMCHECK}_i > \text{FAILDECIDE}_i$, then status of sensor_i is declared failed.

SCORESHEET for Sensor i	
Sensor Data Test Summary	a_i
Filter Data Test Summary	b_i
BIT Indicator Test Summary	c_i
Auxiliary Isolation Test (if any)	d_i
SUMCHECK	$a_i + b_i + c_i + d_i$

$$a_i = \begin{cases} \text{0—no failure indicated} \\ \text{1—one failure vote received} \\ \text{2—more than one, but less than RAWTHRESH}_i\text{ failure votes} \\ \text{3—more than RAWTHRESH}_i\text{ failure votes received.} \end{cases}$$

$$b_i = \begin{cases} \text{0—no failure indicated} \\ \text{1—one failure vote received} \\ \text{2—more than one, but less than FILTRTHRESH}_i\text{ failure votes} \\ \text{3—more than FILTRTHRESH}_i\text{ failure votes received.} \end{cases}$$

$$c_i = \begin{cases} \text{0—no BIT failures indicated} \\ \text{1—one BIT failure registered} \\ \text{2—more than one, but less than BITTHRESH}_i\text{ votes} \\ \text{3—more than BITTHRESH}_i\text{ votes received.} \end{cases}$$

$$d_i = \begin{cases} \text{0—no additional clues that sensor}_i\text{ has failed (or no other test available)} \\ \text{1—weak additional evidence that sensor}_i\text{ has failed (much less than AUXTHRESH}_i\text{)} \\ \text{2—marginal evidence that sensor}_i\text{ has failed} \\ \text{3—overwhelming evidence that sensor}_i\text{ has failed (greater than AUXTHRESH}_i\text{).} \end{cases}$$

Fig. 5. SCORESHEET summarizes results of all FDI tests for assessment by FDI decision function.

In this way, the FAILDECIDE_i thresholds can be simply tailored based on experience to be unique to a particular sensor along with the tailoring accessible through setting of corresponding voting, filtering, BIT test, and auxiliary test thresholds indicated to be available in Fig. 5. Alternatively, all the thresholds can be made identical across the board for all participating sensors for a truly democratic decision. Thresholds can also be relaxed in a prescribed fashion as the conclusion of a mission draws near where navigation accuracy may be of less criticality if that appears to be an appropriate course of action for the specific application.

Human overrides, as called for in [182], can be accommodated as mere changes in the fields a_i, b_i, c_i, and d_i (or as changes directly to SUMCHECK to either make it zero or to be in excess of its threshold) to signal exclusion of this sensor subsystem from further active SMN participation. As long as voting/monitoring continues to be performed on the subsystem and fair democratic practices are invoked, a sensor previously declared as useless or improperly performing (as a consequence of maneuvers or jamming) may return to service when all SCORESHEET SUMCHECK indications are that it is no longer degraded.

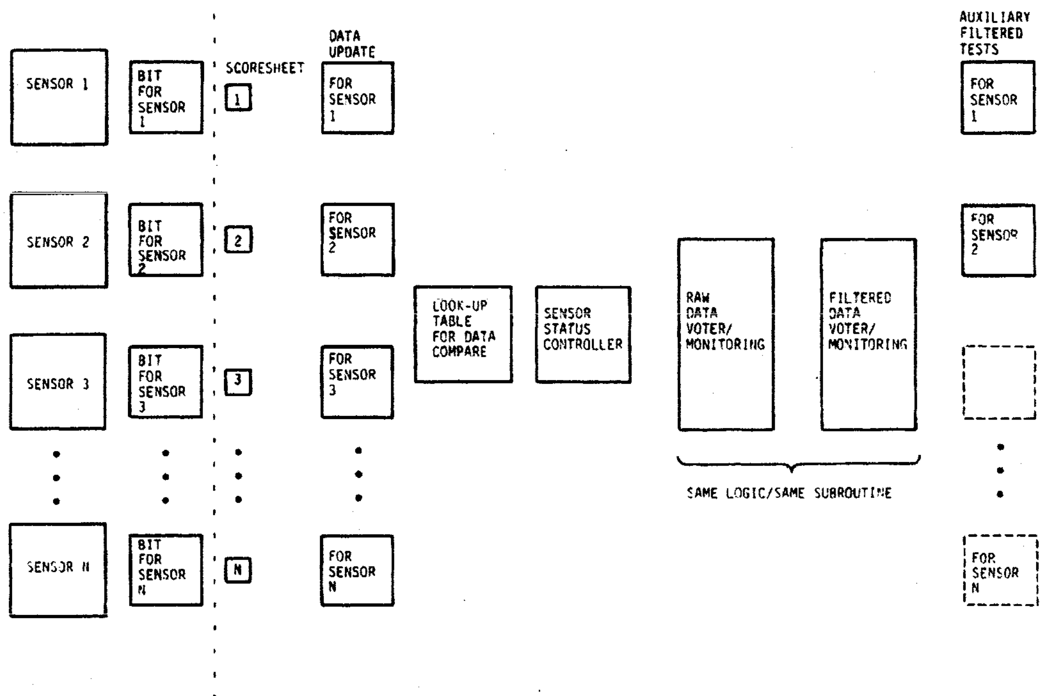
From an analytic viewpoint, each field of the SCORESHEET is derived from a decision test and has quantifiable statistics of false alarm and correct detection. Since the action of forming SUMCHECK is just and "*o*ring" operation of four random variables, each with a certain probability of exceeding its "testbench fine-tuned or experience-derived field-tailored" threshold, it is analytically tractable to specify the FAILDECIDE_i threshold to exhibit specified tolerable false alarm characteristics using standard statistical techniques [146, 193, 194; 177, 178, 179]). Moreover, such performance statistics can be utilized in standard approaches [30] of applying discrete-state Markov analysis of the availability/reliability of these reconfigurable systems as demonstrated in [16, 79]. Recent trends to introduce semi-Markov analysis techniques into this performance evaluation role are currently being developed but appear to require exorbitantly more states (i.e., "the state proliferation problem") and so are less tractable. Recommendations in [151] to prune the state trellis of those states as the primary way to achieve tractable calculations corresponds to not acknowledging the possibility of the occurrence of multiple simultaneous system failures but are unfortunately inconsistent with the realities of the bullets and scrapnell on a battlefield.

One final issue in this voter/monitor design is now raised in order that it can be defused. So far the voting strategy described above has been posed with strict covariance weighting, while in reality certain sensors are known to be considerably more accurate (in a nominally nonhostile benign regime and environment). Thus certain outcomes should apparently be unquestioningly relied upon more readily than on the secondary sensor subsystems. However, a preferred sensor usage hierarchy can in fact still be conveniently superimposed (or even superceded by pilot preference [182]) without altering the architecture of this voter/monitor methodology but by handling these quirks or deviations in standard procedure by mere minor logical operations on the contents of the sensor SCORESHEET data blocks. The desired consequence then occurs as an outcome in the normal operation of the sensor status controller. However, the desirable quality of democratic voter/monitoring is in how it reacts appropriately to bump a particular primary sensor when it is susceptible to a particularly hostile threat environment, yet allows that sensor to resume its lead role when the environmental perturbations subside. Such were the issues that determined this design.

B. Assessing the Computer Burden of the Candidate Voter-Based SMN FDIR Design

An overview of the software modules existing in the FDIR design offered here for SMN is provided in the upper portion of Fig. 6. On the left-hand side are the N sensors available (as the particular version of the SMN set is outfitted with). On the right-hand side are the blocks and boxes representating elements of the SMN FDIR design. The design is depicted in this way to emphasize which software component storage areas have as many instantiations as there are sensor subsystems and which are to be handled as a single subroutine. In this way, the tallies of memory at the bottom of Fig. 6 can be conveniently portrayed as being a factor of N for those items that are replicated N times. Notice that only relatively small and manageable components of only 11 or 28 words are repeated N times. The largest FDIR entity being the voter/monitor is conservatively less than 1000 computer words (of assembly language instructions) and is proposed to be used for both raw data and filtered data voter/monitoring as a common software subroutine slated for dual use.

The operation tallies for six of the seven software components of this SMN FDIR design are depicted at the bottom of Fig. 6. The SENSOR SCORESHEET and the DATA UPDATE areas are not operations but are essentially data buffers operated upon by the other software routines. The table look-up to enable a quick assessment of what data to compare between two sensors should have $N!/(N-2)!2!$ entries representing possible comparisons between N sensor subsystems "taken 2 at a time" (where the case of $N=9$ yields a maximum of



MEMORY TALLY (WORDS)	11N	28N	144	12	844	Low, but not caluclated here
OPERATIONS TALLY (NUMBER OF OPERATIONS)	-	-	$\frac{N!}{(N-2)!2!}$	12	$N^3+187N^2-85N+140+N$ (2X2 MATRIX INVERSION)	Low, but not calculated here

185703-3

Fig. 6. Computational burden of FDIR.

only 36 simple operations). With this design, there is no requirement for dynamically allocated memory in a real-time SMN application, since it is sufficient to set this parameter beforehand corresponding to the total number of subsystems the SMN system is responsible for. In a very worst case situation, this sorting could occur N times for a total of N^3 operations (where the sort is assumed to take N^2 operations for the list of length N). This is an extremely conservative worst case type of bound for sorting operations since recent technological breakthroughs in this area indicate that sorting can be accomplished *almost instantaneously* when parallel processing implementations are used [168–170]. It is very unlikely that the sort will be used N times but prudence in a worst case analysis dictates that this appear in the simple conservative tally of operations counts for the voter/monitor offered in Fig. 6. In practice, the nominal case of no dissenter or no culprit subsystem being encountered requires no sorts, one culprit being implicated involves one sort, etc.

AUXILIARY FILTERED TESTS for refined isolation are prescribed only for certain sensors such as in checking smoothed INS residuals for the presence of an unexpected bias trend of unacceptable magnitude, or in decomposing GPS position error along the lines-of-sight of the four satellites being utilized to reveal when all error is essentially lumped as being contributed by one satellite as an indication of a faulty satellite. These tests could be initiated only when the SENSOR STATUS CONTROLLER declares a sensor to be SUSPECT thus avoiding these computations when such tests are not indicated as needed in the nominal unfailed situation. The logic of these AUXILIARY FILTERED TESTS is straightforward and manageable and is just a slight complication over what is already to be done in the normal filtering mode. Since those tests typically require more time to reach final conclusions by deductive reasoning while considering all possible culprits or sources of test contamination or by requiring more time to allow enough measured evidence to be amassed, AUXILIARY FILTERED TESTS are excellent candidates to be implemented as AI/expert system "inference engines." Such endeavors are to be typically done in LISP on "flyable" symbolics machines (with navigation application precedents being already accomplished in [158]). Thus a lucrative and natural partitioning and handling of the processing burden of the proposed FDI algorithm suggests itself: an AI implementation of the AUXILIARY TESTS for decisions which are not expected to be immediately forthcoming [196], while the major portion of the algorithm can be routinely implemented (without any AI) using conventional techniques which allow real-time FDI indications. The AI-based AUXILIARY FILTERED TESTS could be completed as background tasks and, when ready, its decisions could be incorporated within the architectural framework already described here (cf. [217]). This is a low-risk approach to FDI for SMN that avoids relying entirely on AI algorithms that may currently be incapable of making real-time decisions in the millisecond time frame needed by aircraft. Although not quantitatively tallied here, the computational burden of the AUXILIARY FILTERED TESTS are not anticipated to contain any surprises that would tilt the balance already expressed of the general utility of the FDIR design described here.

IV. ENTRY OF THE ERA OF DECENTRALIZED KALMAN FILTERING

A. A Simplified Tradeoff Analysis Characterizing Use of a Representative Decentralized Filter

Standard techniques for quantifying the computer burden associated with implementing alternative filter mechanizations have been refined over the years (e.g., [120, 121, and 106, ch. 7]) and typically involve assessments of algorithm operation counts, the corresponding algorithm cycle times as determined for the target machine, and allotments of program and stored memory. Precedence in applying such tallies are offered in [97] for several alternative decentralized filtering formulations. The utility of such considerations is quantitatively illustrated below in arguing the case favoring implementing a Kalman filter in distributed form (as with the decentralized reformulation in [97] of the Bar-Itzhack algorithm [90]) on two (or more) processors rather than as a standard single large filter on one processor (that is more susceptible to being throughput limited). The approximations of [135] are a historical filtering approach to navigation data compression and simplification to conserve on-line computer resources as its rationale. Representative filter state selection for SMN type applications are as summarized in Table III.

As an example, specifications for the phase I integration of the JTIDS RelNav and global positioning system (GPS) on the F-16A originally called for utilization of three separate filters, one for GPS, one for JTIDS RelNav, and one dedicated to aided inertial navigation. This type of situation appears a likely candidate for the decentralized B–I multirate filtering approach of [97] as illustrated on the left in Fig. 7. The GPS filter could be used to incorporate position and velocity information at a fast rate in an unjammed environment, then fed to a slower-rate higher fidelity navigation filter used for aiding the inertial navigation system in an integrated manner.

For two separate GPS and JTIDS filters of dimension 12 and 15, respectively, as considered in [122] (which, unfortunately, ignored filter throughput considerations) the advantage of two over one larger 19 state unified filter is obtained from the ratio of the total number of required operations [120] as

$$\frac{(12)^3 + (15)^3}{(19)^3} = \frac{5103}{6859} = 0.74 \tag{3}$$

or a 26 percent reduction in the total number of

TABLE III
Overview Summarizing Previously Suggested Filter Models for GPS, GPS/INS, JTIDS/INS, JTIDS, and GPS/JTIDS/INS

FILTER STATE DESIGNATIONS	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10	No. 11		No. 12	
	GPS (Unaided) Draper 1978	GPS Aided by IMU Draper 1978	GPS Aided by INS Kriegsman & Stonestreet	GPS Aided by INS Widnall Gobbini 1980	JTIDS Nome Stenbaugh Kleinman	JTIDS/INS Draper & Kriegsman & Stonestreet	JTIDS/INS ADM Singer Kearfott	JTIDS/INS MIT Hughes	JTIDS/INS Widnall & Gobbini	JTIDS Unaided* Fried	GPS/JTIDS/INS Widnall & Gobbini	GPS/JTIDS/INS Draper Kriegsman Stonestreet	GPS/JTIDS/INS TRW NFBARS	
Position			Error 3	Error 3		Error 2	Whole Value 2	Error 2	Error 2	Error 3	Error 3	Error 3	3	
Velocity Errors	3	3	3	3		2	2	2	2	3	3	3	3	
Acceleration Errors	3									3				
Angle Misalignments Between Computed and True Tangent Plane Axes		2												
Altitude Error (open-loop)		1					1	1						
External Altitude Reference Error		1	1	1		1					1	1	1	
GPS Clock Phase Error		1	1	1							1	1		
GPS Clock Frequency Error		1	1	1							1	1		
IMU Misalignment Angles		3	3	3		3	3	3	3		3	3	3	
IMU Bias Gryo Drift-Rates								3					3	
JTIDS Clock Phase Error					1	1	1	1	1	1	1	1	1	
JTIDS Clock Frequency Error					1	1	1	1	1	1	1	1	1	
Relative Position Error					2	2	2	2	2		2	2	2	
Relative Velocity Error					2		2	2	2		2		2	
Relative Azimuth Error (with respect to north)						1	1	1	1		1	1	1	
Controller Clock Phase Error											1		1	
Controller Clock Frequency Error											1		1	
Controller Related Velocity Errors												2		
GPS/JTIDS Clock Phase Error														Suggested but not yet used.
GPS/JTIDS Clock Frequency Error														
Correlated Velocity Errors						2								
Heading Error					1									
Heading Drift-Rate Error					1									
Total Number of States	6	12	12	12	8	15	15	18	14	11	21	19	22	

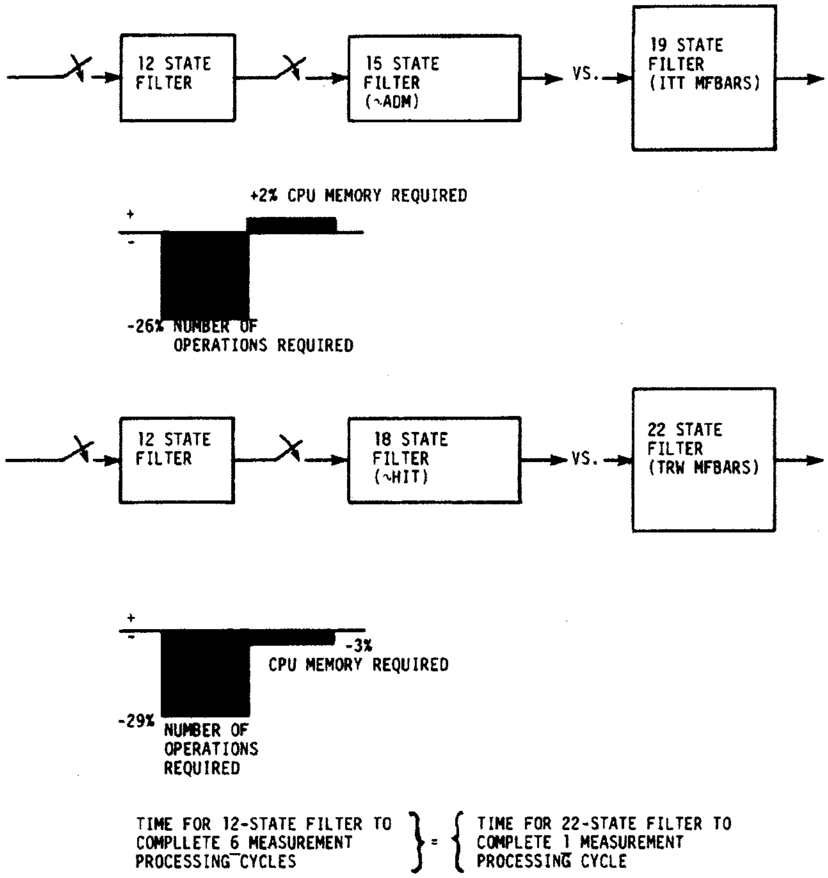


Fig. 7. Benefits of two filters over one federated filter for GPS/JTIDS/INS.

operations to be performed during each filter cycle even though the INS gyro drift-rate states are modeled twice. Unfortunately, a slight 2 percent increase in required computer memory allottment for data is indicated by

$$\frac{(12)^2 + (15)^2}{(19)^2} = \frac{369}{361} = 1.02. \tag{4}$$

However, the large benefit in throughput as the major consideration in such applications appears to be well worth the slight penalty.

The case favoring two separate filters is even more pronounced when considering an alternative state selection [123] corresponding to two filters of state size 12 and 18 versus a single 22 state filter since calculations of the above form indicate *savings* to be achieved in both the number of operations (equivalent to algorithm cycle time of processing a filter measurement) and computer data memory required as, respectively, 30 percent and 3 percent.

If two separate digital processors are used, parallel processing of each of the two filters on different machines provides the advantage that the system is only limited by the slower speed of the single larger filter (of 15 or 18 states). In comparison, the smaller filter of 12 states can proceed through 6 Kalman filter measurement processing cycles in the same time that a larger unified 22 state filter could complete only one cycle, as indicated by the following ratios:

$$\frac{(22)^3}{(12)^3} = \frac{10648}{1728} = 6.16. \tag{5}$$

The conclusion is that a unified single filter will limit processing throughput and hinder full utilization of the GPS measurements available in an unjammed environment. The above arguments are graphically, illustrated in Fig. 7.

The novel original reformulation in [97, sec. 1.5] of the B–I algorithm [90] to now be implementable on two processors as depicted in functional detail in Fig. 8 may be roughly represented as the top diagram of Fig. 9. A generalization to three concatenated but nested filters (i.e., the consecutive filter models are nested by perhaps an allowable similarly transformation) operating at decreasing sampling rates of fast/medium/slow is represented as the middle diagram of Fig. 9. The approach for accomplishing this task is roughly merely a back-to-back repetition of the two filter technique already worked out in [97] but applied separately to each two-filter pair of, first, the fast/medium rate filters, then to the medium/slow filter pair.

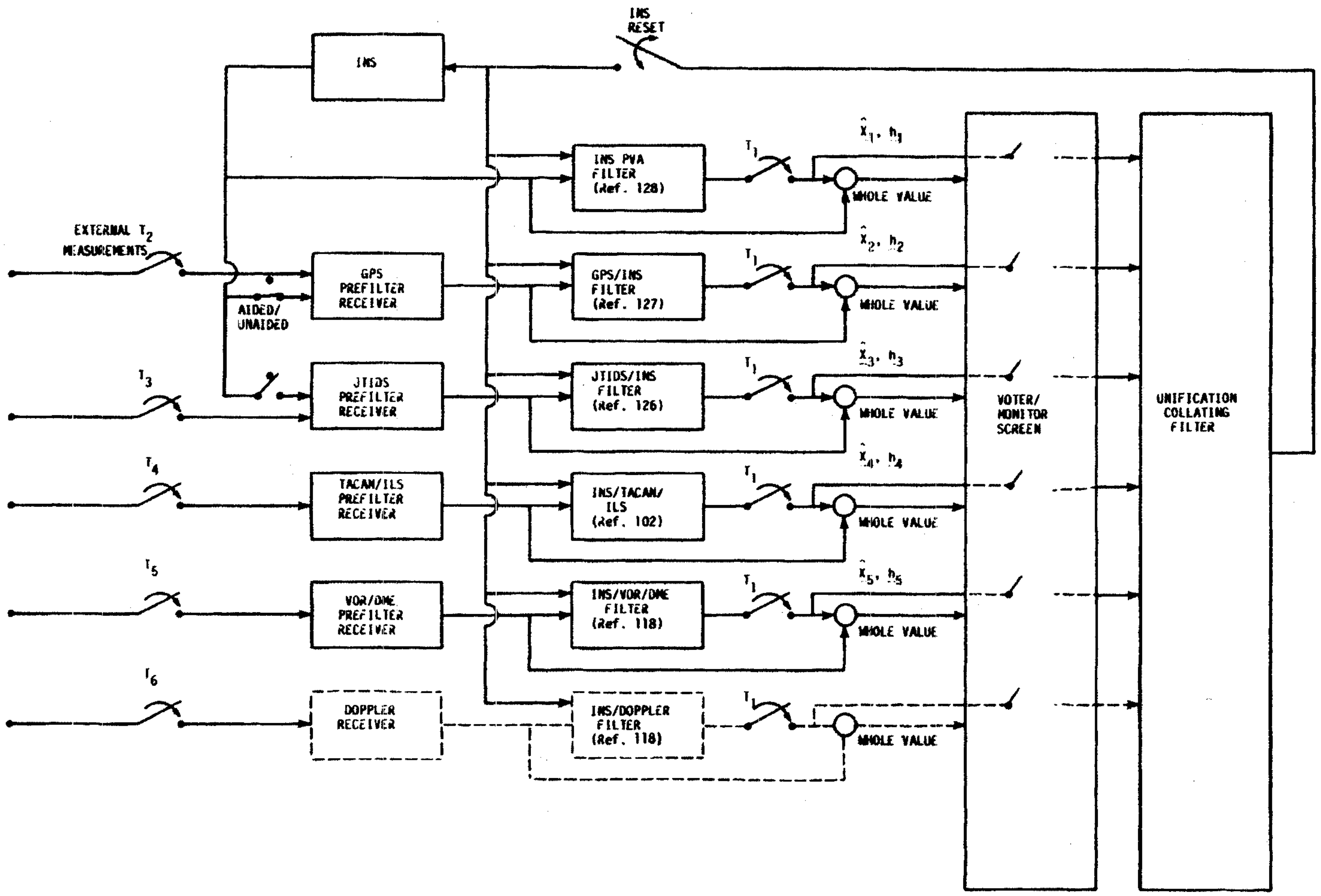


Fig. 8. Decentralized SMN filter design to enhance failure detection/isolation and to ease reconfiguration.

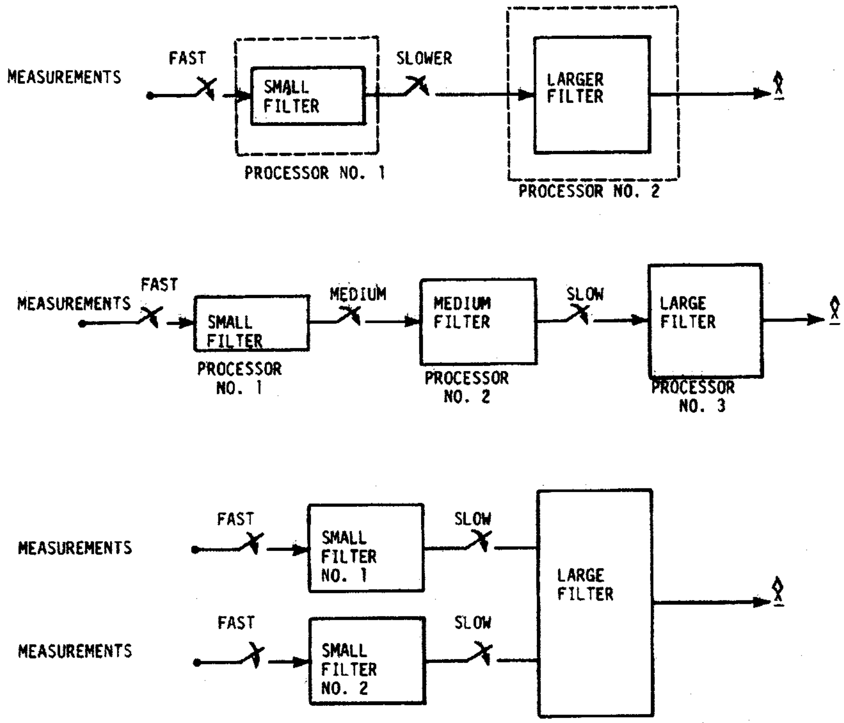


Fig. 9. New approaches to filtering via parallel processing.

B. A Novel Design for SMN Combining Decentralized Filters and Voting

Several aspects of the decentralized implementation depicted in Fig. 1 for the SMN application are discussed now. A practically implemented reduced-order Kalman filter provides a well-understood framework for blending data from a variety of sources to provide variations in performance ranging from reasonably good to merely adequate estimates of the underlying navigation error states. The departure from ideally optimal to merely adequate performance of *any* practical reduced-order filter is a necessary consequence of the tradeoff that is routinely made in exchanging maximum achievable accuracy in estimation for a reduced computational burden in order to allow a real-time mechanization (with constraints on the available memory and cycle time available). Less well known is that a recent decentralized filter implementation [86, 87] is available that can *also* reconstruct fully optimal state estimates from data passed to it from several participating local filters designed to accommodate a particular sensor with *not* necessarily identical measurements or measurement types from these sensors (i.e., differing observation matrices and measurement noise covariance intensity levels) but with a degree of commonality in the underlying states of interest (with possibly differing models for these states[1] as different levels of fidelity for the various local filters). A departure from the full optimality of a global estimate is also possible for this implementation if any of the participating subsystem filters is of reduced order, but the degradation incurred is comparable to what is usually tolerably incurred in using a reduced-order centralized filter.

The Speyer version of a decentralized Kalman filter, as represented in Fig. 1 (but without the voter/monitoring screen), collates all the estimates and auxiliary data passed to it into a single unified global estimate. The inclusion of a voter/monitor screen is an *original contribution* introduced here that is especially compatible with the structure of the Speyer version of decentralized filtering. This design readily handles all combinations of actively participating sensors and the associated subsystem filters, thus providing the best single solution from the available data. The necessary interconnections between filters (hosted on their individual processors) could be made via databuses ([220, 221]). (It thus avoids the problem of ad hoc selection of a single solution from several individual solutions of differing accuracies since no such ambiguity exists in the formulation of [86, 87]). Use of the voter/monitoring screen, as discussed in the appendix (with cross-state comparisons depicted in Table II), for initial weeding from contributing filters of failed sensors prior to collation is a new variation unique to the original design portrayed in Fig. 1 for the SMN application.

Consider the following itemized perspectives on the applicability and advantages of using the decentralized implementation depicted in Fig. 9 for the SMN application.

[1]Details of how to handle several local filters with variations in underlying truth model are explained in [88, 89, 104].

(1) This decentralized filter makes use of standard models for the individual navigation subsystems—INS, GPS, JTIDS, etc.; models are individualized, yet the solution is integrated.

(2) The Speyer version of decentralized filtering, depicted in Fig. 1, provides a single, self-consistent source of INS feedback corrections. The reset is also conveyed in the manner depicted in the figure to each of the participating filters (including a model for the INS) as being merely a deterministic control, as is standard practice [106].

(3) As depicted in Fig. 1, this decentralized filter provides a single, self-consistent source of receiver rate-aiding corrections, even if one or more sensors are unavailable (e.g., GPS being jammed).

(4) This filter can use any preexisting validated source selection logic already tailored to the particular subsystem at preexisting measurement rates T_2, T_3, T_4, T_5, T_6, T_7 as arrived at by other investigators. Such inherited source selection logic is usually of known bounded complexity (code length, memory required, and processing delay incurred) with previously established adequacy.

(5) The Speyer version of decentralized filtering (or its variations as in [88, 89]) requires fairly little in the way of new code builds since it can make use of the existing validated local filters. Only the unification collating filter need be built as a new initiative. Some slight modification in the form of a low-order augmentation is needed for each of the participating local filters to include a calculation of the auxiliary variable $h_j(t)$ as well as incorporating a mechanism for communicating both the vector $P^{-1}{}_j x_j$ (of dimension n_j) and the auxiliary variable h_j (of dimension n_j) to the unification collating filter. (The Speyer filter recognizes that it is unnecessary to convey the entire covariance information from any local filter to the collating filter in order for a globally optimum estimate to still be produced.) A mechanism for conveying INS resets to each local filter also modeling INS effects is necessary, but simply achieved as already described under item (2).

(6) This version of decentralized filtering is apparently fairly inexpensive to build since it is to make extensive use of any preexisting component sensor subsystem filter designs with only minor modifications. However, as discussed above in item (5), the unification collating filter would be a new initiative.

(7) As outgrowths of items (1) and (5) above, this version of decentralized filtering *should* be able to use off-the-shelf preexisting filter implementations for the local subsystem filters (with perhaps slight modifications) such as those provided as (a) GPS/INS—Rockwell Collins or Magnavox (as summarized in [105, 143–145, 184, 185]); JTIDS/INS—HIT (as summarized in [116]); INS/TACAN/ILS [83]; INS/VOR/DME [99]; INS/Doppler radar [106, p. 304]; INS/SAR/Baro-altimeter/Doppler radar [199], [200], [240].

(8) The *primary motivation* for considering the design of Fig. 1, utilizing decentralized filters in the manner depicted, is the *protection that it provides when/if a subsystem fails*. Once declared to be failed or degraded by the sensor status controller (Section IIIA), the culprit system can be prevented from further immediate participation in the global estimate merely by diverting its estimates and associated auxiliary vector from inclusion in the inputs of the unification collating filter. In this way, the unification collating filter is allowed to do the best that it can with what it has available. Another major consideration favoring this approach involving a decentralized filtering implementation is how it facilitates voting on filter-corrected whole value estimates (along with their mitigating covariances that reflect the realities of the current circumstances[2]) as a primary method for detecting and isolating failures.

(9) It is no longer necessary that the two or more filters run at the same rate (but the local filters should be synchronized to integer multiples of a common sample rate T_1 so that cross-comparisons between estimates can be conveniently accomplished without worry of unacceptable senescence or computationally burdensome time-tagging and extrapolation to a common cross-check time).

(10) Whereas a failure in *any* subsystem can contaminate the single "best" solution of a totally unified central filter-based design and can also adversely affect data associated with other subsystems as perceived by a central filter, this problem is avoided entirely with a decentralized implementation since the voter/monitoring and isolation takes place prior to completion of this fundamental step of unifying estimates, thus allowing extraction of any information from failed or degraded subsystems prior to incorporation of remaining good information. *Another factor is that additional auxiliary tests for failure signature monitoring on the estimates associated with a particular sensor subsystem can now be tailored explicitly to the specific sensor by being attached to (or performed on) the outputs of the local filters* for that sensor without worry of significant cross-contamination that would likely cloud such tests if they were applied to a unified filter. Additionally, all participating local filters are stand-alone so that unwieldy transients are avoided in adding or removing a subsystem sensor from participation without altering the overall controlling superstructure.

As an adjunct to the comments offered in above items (4), (6), (7) on carry-over of the work and experience of the prior designers of any preexisting local filters. These designers of the local filters have already adequately matched the operational constraints on each subsystem and have met its operational goals and specifications (such as way-point navigation or area navigation) [239].

[2]Larger covariances for INS and GPS during high-g maneuvers, larger covariances for JTIDS and GPS in the presence of jamming or bad geometry, etc.

Adapting these designs to SMN should be relatively easy since the computer capacity for imbedded implementation is likely to be much greater in current and future SMN applications though perhaps less dedicated to the specific local filter. Still, the original filter implementations in general must be converted to a common computer language such as Ada [163] or made compatible with the operating system used for SMN; however, there is no need to duplicate the effort in obtaining the underlying descriptive equations other than perhaps changing the step size and making modifications in response to this change of well-known consequences (such as in changing the intensity of the process noise covariance level or in altering some of the terms in the transition matrix).

The major resistance against use of the decentralized filter design of Fig. 1 stems from the fact that there have been few known precedents for its use (successful or otherwise) in particular navigation applications and so it is usually perceived by many as being a high-risk item. However, the suboptimal filtering approach of Bar-Itzhack [90–96] was offered in a decentralized filter framework in [97] useful for INS/GPS/JTIDS applications in Fig. 8. A similar two-filter form for a GPS/INS application is also currently being applied as discussed in [98] and, independently, studied in other GPS/INS contexts in [152, 197]. JPL is currently pursuing a square-root formulation of a generalization of [88] as [112, 113, 153] for a tracking application of satellite position determination. Additional decentralized filtering approaches have recently emerged as [154, 156, 198, 222, 223] and on pp. 113–115 of [233].

Dosh and Yakos [152] ostensibly took a two-Kalman-filter approach to GPS/INS integration because it provides a "transparent" design that simplifies the handling of both state 5 and state 3 GPS receiver operation. Some apprehension is expressed here concerning many aspects of the intermediate and final conclusions in the study of [152] for the following reasons.

(1) Several different interpretations can be ascribed to the somewhat sparse discussion in [152] concerning a claim that the measurement residual being not yet filtered by the GPS filter at the time of its transfer to the second filter, and the subsequent claim that this residual thus "has the attribute of raw measurement data (i.e., white noise measurement error)." A clarification on this issue is needed since [152] invokes this aspect as being the key that supposedly justifies their further claim that the "filter in the central computer can use this residual data and provide *independent* filtering." Juxtaposition of both filter residuals apparently contradicts the claim in [152] of having independent filter performance since there is evidently strong correspondence of events in time where simultaneous mutual spikes occur even if sometimes of opposite polarity.

(2) By employing this "residual data interface" between the two filters, Dosh and Yakos [152] claimed that this aspect allowed a single-channel GPS set to be used for this GPS/INS integration test of concept. Further explanation is needed to clarify just how this accommodated use of a single-channel GPS set.

(3) Dosh and Yakos [152] never identified the states being utilized in their 20-state second filter as is necessary for an understanding or repetition of the experiment.

(4) Simulated laboratory testing followed by subsequent van tests were indicated and results were shown; yet nowhere were prevailing conditions cited (such as the current GPS State, GDOP, C/No, FOM, *true* user position and velocity) that are crucial for a fair evaluation.

(5) Juxtaposition of velocity estimates from both filters still reveals nothing about adequacy of filter performance since proximity to true velocity is obscured.

(6) Unfortunately, Dosh and Yakos [152] apparently present no clear evidence that this two-filter design "has eliminated the risk of instabilities associated with using fitler data" as is claimed.

Somewhat surprisingly, decentralized filters have been successfully used for over ten years in navigation applications in C-4 Trident submarines and C-4 backfit Poseidon submarines (see [16, 218, 219] and [212, sec. 6]). However, no coherent decentralized filter theory for coordinating the operations of the two separate filters (SINS correction filter and ESGM reset filter) was ever invoked. Instead, each filter was treated as if it were by itself a unified single filter and standard Monte Carlo and covariance analysis were used for performance predictions and validation. More efficient submarine navigation could probably be realized if the true decentralized nature of these C-4 Trident filters were acknowledged and a rigorous decentralized filter formulation were invoked to support the analyses for a more exacting evaluation that matches "apples to apples."

A further precedence has now been encountered in successful use of two separate Kalman filters in [157]. SITAN was developed by Sandia Corporation in 1983 as an alternative terrain correlation position fix source to TERrain COntour Matching (TERCOM). An aircraft using SITAN:

- can be almost continuously refined with position information obtained from its radar updates through comparisons with what is available from its stored map;
- has greater flexibility in its allowable maneuvers because it is not constrained to specific discretely located presurveyed and preplanned fix sites, as TERCOM is;
- can use much lower power radar fixes (less detectable by an enemy's surveillance or homing weapons) since SITAN again does not rely on a sparse number of "do-or-die" radar fix site locations.

SITAN allows for extensive use of an INS and in so doing utilizes two different Kalman filters. TASC's 1985 phase I AI approach [158] to an advanced tactical navigator (ATN) for WPAFB involved simulation demonstrations on an interconnected Symbolics/VAX/IBM computer hook-up and notably also used two Kalman filters in its emulation of SITAN usage.

C. Analytical Aspects of the Decentralized Filter Design for SMN

Given several redundant measurement sensors of the following form:

$$z_j(j) = H_j(k)x(k) + v_j(k), \quad \text{for } j = 1, 2, \ldots, M \tag{6}$$

(where H_j, v_j are the observation matrix and uncorrelated white Gaussian measurement noises, respectively, and x and z_j are the global system state and measurements at the sensor, respectively), it is reasonably well known [107] that the optimal linear least mean square estimate of $x(k)$ has the form

$$\hat{x}(k|k) = \left[P^{-1}(k|k-1) + \sum_{j=1}^{M} H_j^{\mathrm{T}}(k)R_j^{-1}(k)H_j(k)\right]^{-1} \cdot \left[\sum_{i=1}^{M} H_i^{\mathrm{T}}(k)R_i^{-1}(k)z_i(k) + P^{-1}(k|k-1)\hat{x}(k|k-1)\right] \tag{7}$$

with associated covariance of estimation error provided by

$$P(k|k) = \left[P^{-1}(k|k-1) + \sum_{j=1}^{M} H_j^{\mathrm{T}}(k)R_j^{-1}(k)H_j(k)\right]^{-1}$$

$$P(k|k-1) = \Phi(k,k-1)\,P(k-1|k-1)\,\Phi^{\mathrm{T}}(k,k-1) + Q_k \tag{8}$$

where $P(k|k-1)$ is the covariance of error of estimating x at time step k, as propagated from $k-1$ for the full state aggregate. Φ is the transition matrix, R_j is the covariance of the additive measurement noise at sensor j, and Q_k is the covariance of the system process noise at time k. Speyer's filter [86, 87] is equivalent to the following form:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \sum_{j=1}^{M(k)} K_j(k) \times [z_j(k) - H_j(k)\hat{x}(k|k-1)] \tag{9a}$$

$$= \hat{x}(k|k-1) + \sum_{j=1}^{M(k)} P(k|k)H_j^{\mathrm{T}}R_j^{-1} \times [z_j - H_j\hat{x}(k|k-1)] \tag{9b}$$

$$= \hat{x}(k|k-1) + P(k|k)\sum_{j=1}^{M(k)} H_j^{\mathrm{T}}R_j^{-1} \times [z_j - H_j\hat{x}(k|k-1)] \tag{9c}$$

but where, instead of as in [107], *several decentralized local estimators* $\hat{x}_j$ are used in the mechanization as occurs in

$$\hat{x}(k|k) = \sum_{j=1}^{M(k)} \{P(k|k)\,[P_j^{-1}(k|k)\hat{x}_j(k|k)] + h_j(k)\} \tag{10a}$$

$$= P(k|k)\sum_{j=1}^{M(k)} P_j^{-1}(k|k)\hat{x}_j(k|k) + \sum_{j=1}^{M(k)} h_j(k) \tag{10b}$$

where estimation error over restricted subsets of the measurement data (as considered in [108]) are defined as

$$P_j(k|k) = E[(x(k) - \hat{x}_j(k))\,(x(k) - \hat{x}_j(k))^{\mathrm{T}}|Z_j(k)]$$

and each $h_j(k)$ satisfies a recursive equation of the form:

$$h_j(k) = F(k)h_j(k-1) + G_j(k)(z_j(k) - H_j(k)\,\hat{x}_j(k|k-1)) \tag{11}$$

where $F(k)$ and $G_j(k)$ are matrices calculated on-line as specified by

$$F(k) = P(k|k)\,[\Phi(k,k-1)\,P(k-1|k-1) \times \Phi^{\mathrm{T}}(k,k-1) + Q(k-1)]^{-1} \times \Phi(k,k-1) \tag{12}$$

$$G_j(k) = F(k)P(k-1|k-1)P_j^{-1}(k-1|k-1) \times \Phi^{-1}(k,k-1) - P(k|k) \times [\Phi P_j\,(k-1|k-1) \times \Phi^{\mathrm{T}} + Q(k-1)]^{-1} \tag{13}$$

and the local subsystem filter is specified as

$$\hat{x}_j(k|k) = \hat{x}_j(k|k-1) + P_j(k|k)H_j^{\mathrm{T}}R_j^{-1} \times [(z_j(k) - H_j\hat{x}(k|k-1)]. \tag{14}$$

D. Voting on Filtered Estimates

By preventing such fairly drastic on-off switching from being used to completely remove the apparently defective component from the loop, the so-called dissenting component may be utilized later if/when it comes back into reasonable agreement (or when some other component is even more degraded than the original dissenting component), so that the original dissenter may then be used as a fall-back support measure for replacement or augmentation of a component that is worst off. Such flexible practices are especially important in situations where some components are more sensitive than others or with performance that is more adversely affected than others in specific regimes or environments. Representative examples of where this would be an important consideration in military related SMN applications are

- during high-g maneuvers (affecting conventional gimbaled gyros and GPS aided/unaided);
- close hostile high powered or sophisticated agile jammers (affecting radio-based systems);
- during significant interference (affecting radio-based systems);
- during receive-only operations (as can occur for a passive JTIDS user);

- in situations of unfavorable geometry (e.g., bad GDOP for GPS; low observability, low relative motion, or bad orientation with respect to two navigation Controllers for a JTIDS user).

Thus, the total system benefits from complementary synergistic characteristics when availed (as with SMN) with several simultaneous sources of navigation information, even though it is of varying quality (that is essentially known or may be reasonably ranked on-line by estimation). All these considerations are of concern in FDIR for overall navigation system redundancy management.

E. Sensor Interface Handlers

Requirements exist for properly passing data from one subsystem to another and for converting data obtained from certain sensor subsystems into a desired form for use either in the integrated navigation function or in the FDIR function for SMN. For the INS, there can be requirements on passing VOR/DME data to the INS [111] as well as baroaltimeter data that is the usual fall-back for INS vertical loop damping. When good stable GPS altitude data is available, it has been historically preferred as the source of damping of the effective vertical channel. Baro-altimeter data is also used as a secondary reference in Doppler AHRS. Baro-altimeter data is calibrated to a reference level and altitude conventions differ from that of INS, GPS, and JTIDS. Occasionally INS resets from a Kalman filter are also to be passed across sensor subsystem boundaries and must be appropriately accounted for if conventions differ as to units, coordinate systems, and sampling rates.

While most navigation sensors express final results in a WGS-72 coordinate system, the navigation information provided by such subsystems as PLRS currently express position in MGR coordinates that should be converted to WGS-72 [180] or WGS-84 as now used by Phase III GPS within the interface handler. Similarly, the natural coordinate frame for GPS calculations is ECEF because of calculations that must be performed regarding satellite orbital positions; however, the conversion to WGS-72 [188], [235] is necessary for GPS participation in sensor-to-sensor cross-comparisons within raw data Voter/Monitoring. Similarly, JTIDS and INS results need to be ultimately expressed in WGS-72 coordinates for voting. An ENAC 77-1/SNU 84-1 INS may be either conventionally gimbaled or be strapped down to aircraft body frame coordinates and may therefore utilize supporting quaternion calculations which may need to be backed out [183] in the interface handler prior to participation in voting. Even special purpose alignment/calibration mode calculations may be handled in the interface handler, if necessary.

The key to handling so many functions, within the interface handler, is in using a "smart" interface handler with possibly memory and processor capabilities, where some software code may reside. This was a key feature of the Collins' Phase II GPS design [143, 144, 184, 185] where similar tactics were used in the so-called flexible modular interface (FMI) which therefore allowed commonality among principal equipments (the GPS sets), but handled sensor-to-sensor and sensor-to-host platform issues within the FMI [105]. This design approach offered ease in accommodating a variety of host platforms with differing operational objectives, missions, maneuver capabilities, and operational environments by handling these issues associated with crossing system boundaries within the FMI processing.

V. ASSESSING REMAINING BARRIERS TO ACHIEVING KALMAN "FILTER-ON-A-CHIP"

The possibility of being implemented as a VHSIC chip has been a driving consideration for reformulations, when amenable, of many well-known matrix algorithms that are used in a variety of application areas from radar and communications to navigation. Practical motivation for a reformulation is that algorithms that previously were of the order of n^3 in time, by requiring n^3 operations when implemented on standard Von Neumann sequential machines, can now be reduced to be of order n^2 in time, when the algorithms' structure is such that operations can be distributed in space over n channels so that n simultaneous operations are performed in parallel (and could possibly be ultimately reduced to being merely of order n in time if n^2 operations can be simultaneously performed in parallel over an area–volume). One structure that accommodates such speeded-up processing is a systolic array [134], where the results of computations are passed along in a structured fashion to nearest neighbors as a continuous wavefront of processing to be performed, while the desired answer is pushed out the other end. Another structure that accomplishes faster computations is a specialized CORDIC processor ([129, 130]) that is specially designed for implementing algorithms that primarily involve geometric rotations (e.g., Householder transformations, QR- and LR-algorithms). Use of either of these two processing techniques requires reformulation of the algorithm in general, but may only be accomplished if the original algorithm has a structure that is amenable to such recasting.

It is now evident that all the associated transformations necessary for conveniently implementing a Kalman filter are already available in systolic versions; therefore "Kalman-filtering-on-a-chip" will become a reality soon.

The following filter-related algorithms have been announced as systolic implementations:

(1) Gauss–Jordan reduction (no pivoting);
(2) highly parallel architectures for solving linear equations [131] [241];
(3) bit level systolic arrays such as implementable in the commercially available NCR45CG72 GAPP;
(4) recursive least squares;

(5) recast version of so-designated fast Kalman algorithm of Falconer and Ljung [147];
(6) pure time updates for least squares lattice algorithm [148];
(7) calculation of eigenvalues via QR-algorithm (for symmetric, banded, and general matrices) using systolic arrays [149] [129]–[134];
(8) reduction of matrices to generalized Hessenberg form via systolic arrays [149];
(9) Choleski decomposition of a positive definite matrix into two triangular matrix factors [173];
(10) Householder transformations [232];

and there is an approach being developed for parallel pseudo-random number generation [234] (that could ostensibly speed up Monte Carlo simulations for decentralized filters as addressed at the end of Section VI). These analytic stepping stones are already available to be further assembled to make decentralized "Kalman-filtering-on-a-chip" for SMN applications become a reality. Even pseudo-inverse calculations [187], [242] are now available from systolic arrays via a singualr value decomposition [150], and such simple operations as scalar division have been recast in a systolic form to be as fast as multiplication [210]. A recent occurrence is in the understanding of how several exhaustive alternative systolic realizations can be obtained for the same algorithm via projections [207–209] in seeking a preferred realization. However, caution must be exercised since some of the realizations of [208, 209] are "systolic" only in a broad sense since they can extend beyond just nearest neighbor interaction.

In the past, attempts at pipelining or parallel processing of Kalman filter algorithms were not very successful [229] because they were constrained to use hardware architectures previously developed primarily for FFT calculations, which are structurally different. There are already claims ([126–128]) of current availability of a parallel processor implementation of a Kalman filter. However, the version of [126–128] currently appears to be somewhat contorted from what is classically known as a Kalman filter and is yet (1985) to be decomposed for a systolic or CORDIC implementation.

It is generally known that one major constraint in seeking to achieve *any* algorithm-on-a-chip is to try to enforce that all operations are self-contained on just one chip in order to avoid higher power consumption requirements otherwise needed for outside cross-communication and likely necessity of imposing cross-time-synchronization. However, it has been recently revealed that some algorithms may be implemented in a decentralized uncoordinated fashion as wavefront arrays so that problem pieces are independently asynchronously calculated in a somewhat recursive fashion (with feedback and feed-across being allowed); yet a globally correct final answer emerges from this decentralized partitioning of the computations [236]. This aspect is of interest if several chips must be cross-connected in order to achieve the decentralized filtering goals of SMN.

While radar applications of Kalman filtering are also pursuing VHSIC implementations on systolic arrays, the dimension of the usual filters encountered in the radar application is usually small [138]. Consequently, there is a less impressive benefit in going to decentralized implementations other than being able to allocate many separate small filters for multitarget threat tracking or in having filter chips directly behind the radar elements in large radar arrays to cut down on power transmission requirements. On the other hand, navigation applications typically have enough states to make the gains achieved in decentralized VHSIC/systolic array implementation payoff even in the short term in either reduced computational burden or in increased rapidity of cycle time so that sampling rates can be increased and fresher data utilized, as needed on fast moving platforms. It is hoped that more attention will eventually be paid (than initially offered in early VHSIC pursuits of Kalman filters) to numerically stable filter implementations such as in use of Joseph's form of the covariance update equation or in use of Bierman's UDU^T formulation. With the development of VHSIC Hardware Description Language (VHDL), it should be possible to emulate the performance of software algorithms (and perform investigations of the efficacy of alternative layouts on a VHSIC chip) and assess the beneficial effects on algorithm efficiencies reaped through use of systolic arrays.

VI. SUMMARY

Failure detection techniques compatible for navigation systems were surveyed in Section II, refined for the specific SMN avionics application in Section III, and adjoined with a decentralized filtering formulation in Section IV as a design recommended as being worthy of further consideration.

Recapitulating, there are four significantly desirable properties of the decentralized SMN design offered in Section IV (Fig. 1):

(1) resolving differing periods between measurement updates (as T_2 to T_7, for participating subsystems) is handled automatically without significantly altering underlying currently fixed design of preexisting stand-alone subsystems by merely synchronizing the sampled outputs of each subsystem's existing filter to an integer multiple of a common base period T_1 at which the unification collating filter operates;
(2) if any one or more of the several sensor subsystems fails or goes off-line for any reason (e.g., separate processor failure, jamming interference, satellite distruction, fundamental clock failures, anticipated normal mode hardware failures, unanticipated scrapnell-induced failures, EMP, etc.) this structural framework appears to offer the best estimate with the remaining information still provided;
(3) handles cross-correlation information routinely as imbedded in the computations of Eqs. 11–13;

(4) instead of each local filter transmitting (via [203]) *all* raw measurements $z_j(k)$ to the other subsystems or to the master unification collating filter it is sufficient in the design of [86, 87] to transmit only two n_j-dimensional vectors: $P_j(k)^{-1}x_j(k)$ and $h_j(k)$ since the summarizing quantity $h_j(k)$ is itself a function of the local measurements.

The generalization of the Speyer filter that occurs in [88] is such that the structure of Fig. 1 is still adhered to, but

- individual system models of the subcomponent filters are *not* necessarily identical; yet the unification collating filter still correctly accounts and compensates for such differences in providing optimal estimates;
- global and local models must imply the same physical relationship among the existing measurements (e.g., if two measurements are considered to be independent by the Global model, then no local model is allowed to lump them together and, conversely, if the global model describes a measurement as redundant, the same redundancy must appear in the local models that utilize it);
- Speyer [86] and Chang [87] offer a nonhierarchical design in the relationship between subsystems, while [88] presumes a single central unified filter (exactly consistent with Fig. 1 minus the new voter/monitor);
- the single centralized unification colating filter no longer has to consider each transmitted vector $h_j(k)$ (as required in the designs of [86, 87]), but need only be concerned with a single vector as the sum

$$r(k) = \sum_{j=1}^{M(k)} h_j(k)$$

and $r(k)$ is propagated recursively by

$$r(k+1) = \bar{F}(k)r(k) + \sum_{j=1}^{M(k)} K_j(k)\hat{x}_j(k)$$

where $\bar{F}$ and K_j are defined in [88] but are consistent with the original design of [86, 89]. Of course, considerable simplifications occur in the above formulation of [88] when local models are identical to global models as occurs to an extent in the SMN application.

It is further remarked that since the alternative decentralized filter formulations of [86–89] correspond exactly to the output of a centralized global Kalman filter, theoretical stability is not an issue. Since the outputs are identical, the decentralized implementation is asymptotically exponentially stable as is the global centralized Kalman filter. However, numerical stability still *is* an issue that needs to be addressed further. The descriptive equations presented here and dwelled on in [86–89] emphasize internal system structure rather than the most computationally efficient or most numerically stable implementation of the decentralized filter. A so-designated Bierman-type UDU^T square-root filter formulation and/or an information filter formulation [p. 946, 228], [229] would be more computationally expedient for implementing the decentralized filter by, respectively, providing the requisite stability in recursive computations so important in real-time mechanizations over long mission times and in avoiding unnecessary inverses of covariance matrices. There remain some issues to be clarified or refined relating to the amount of approximation incurred in using multiple sampling/utilization rates and [171] may help in this resolution. The effect of using reduced-order filters (when an acknowledged higher order system truth model is present) can only be evaluated via more expensive Monte Carlo simulations at the present time for most decentralized filter formulations (except for the approximate approach of [90] as discussed in [91–93]). Rigorous covariance analysis evaluation tools for decentralized filters are currently absent and are wanting.

While the CRAY-2 has four processors and 16 million 64-bit word memories [225], the G_AA_S-based CRAY-3 to be available in 1988 will ostensibly have 16 processors and should be able to simultaneously handle simulation of the decentralized filtering approach advocated in Section IVC. The CRAY-3 could handle evaluation of each separate indicated sensory designated filter on a separate processor as well as reserving processors for the screening and collating functions. The trajectory generation functions (with cross-coupled specific forces of banking now realistically handled), the feedback resets of the INS, and failure emulation simultaneously in parallel while yet another processor could emulate the location dependent effects of jamming and wind shading.

APPENDIX
RELATIONSHIP BETWEEN CHI-SQUARED STATISTICS AND WEIGHTED PAIR-WISE CORROBORATION TESTS OF THE VOTER/MONITOR

In order to quantitatively compare consistency between the outputs of several sensor subsystems in pairs-of-two to enable eventual voter/monitoring of the totality, the N available sensor outputs are assumed to be of the following form:

$$w_1(t) \begin{array}{l} \triangleq \text{whole value measurements from sensor 1 at time } t \\ = \text{true value}(t) + x_1(t) \end{array} \tag{A1}$$

$$w_2(t) \begin{array}{l} \triangleq \text{whole value measurements from sensor 2 at time } t \\ = \text{true value}(t) + x_2(t) \end{array} \tag{A2}$$

$$\vdots$$

$$w_N(t) \begin{array}{l} \triangleq \text{whole value measurements from sensor } N \text{ at time } t \\ = \text{true value}(t) + x_N(t) \end{array} \tag{A3}$$

where $x_i(t)$ is the error or deviation of sensor i from the true value at time t and are Gaussianly distributed, zero mean random variables (as per the usual assumption for navigation errors that permits the successful use of Kalman filtering). Reiterating, the voting philosophy currently being recommended is to look at the divergence between dimensionally compatible quantities of the same generic type of the following form:

$$w_1 - w_2(t) = [\text{true value}(t) + x_1(t)] - [\text{true value}(t) + x_2(t)] = [x_1(t) - x_2(t)] \tag{A4}$$

$$w_2 - w_N(t) = [\text{true value}(t) + x_2(t)] - [\text{true value}(t) + x_N(t)] = [x_2(t) - x_N(t)] \tag{A5}$$

$$\vdots$$

$$w_N - w_1(t) = [\text{true value}(t) + x_N(t)] - [\text{true value}(t) + x_1(t)] = [x_N(t) - x_1(t)] \tag{A6}$$

where attention is now focused on the remaining simplified terms within the brackets.

Either from simplified maximum likelihood estimator accuracy assessments derived from simple computations [114] on the raw data (by being striped off of status words of received data messages), or from on-line filter computations, appropriate covariances $P(t)$ for each component subsystem (as influenced by fixes, maneuvers, possible significant latitude changes, effects of jamming, possible bad geometry, etc.) are available from which proper compatible subsets of covariance matrices may be extracted for the SMN application at each check time. These covariance subsets are assumed to be available for the calculations and derivations that are discussed next. Denote the following three $p \times p$ covariance matrix subsets of generically similar quantities as

$$\bar{P}_{12}(t) = \text{covariance of } [x_1(t) - x_2(t) | z(t)] = [P_{11} + P_{22} - P_{12} - P_{21}{}^{\mathrm{T}}] \tag{A7}$$

$$\bar{P}_{2N}(t) = \text{covariance of } [x_2(t) - x_N(t) | z(t)] = [P_{22} + P_{NN} - P_{2N} - P_{N2}^{\mathrm{T}}] \tag{A8}$$

$$\vdots$$

$$\bar{P}_{N1}(t) = \text{covariance of } [x_N(t) - x_1(t) | z(t)] = [P_{NN} + P_{11} - P_{N1} - P_{1N}{}^{\mathrm{T}}] \tag{A9}$$

where $z(t)$ in the above represents the measurements of Eq. 6 up to time, t, and the conditioned expectation is being utilized. Exact on-line computation of covariance matrices of both x_i and $\hat{x}_i$ is as indicated in Eqns. 4, 7, 11, 18]. These submatrices may be utilized as described below.

It is desirable to have compatibility measures that monitor the agreement between the outputs of the various sensor subsystems while also considering the *cross-covariances* (i.e., possibly including an accounting of the cross-correlation or cross-coupling in latitude, longitude, and altitude estimates). Such a measure is provided by

$$l_{12}(t) \triangleq [w_1 - w_2]^{\mathrm{T}} \bar{P}_{12}^{-1} [w_1 - w_2] = [x_1(t) - x_2(t)]^{\mathrm{T}} \bar{P}_{12}^{-1}(t) [x_1(t) - x_2(t)] \tag{A10}$$

$$l_{2N}(t) \triangleq [w_2 - w_N]^{\mathrm{T}} \bar{P}_{2N}^{-1} [w_2 - w_N] = [x_2(t) - x_N(t)]^{\mathrm{T}} \bar{P}_{2N}^{-1}(t) [x_2(t) - x_N(t)] \tag{A11}$$

$$\vdots$$

$$l_{N1}(t) \triangleq [w_N - w_1]^{\mathrm{T}} \bar{P}_{N1}^{-1} [w_N - w_1] = [x_N(t) - x_1(t)]^{\mathrm{T}} \bar{P}_{N1}^{-1}(t) [x_N(t) - x_1(t)] \tag{A12}$$

which has several additional useful properties as well. (Please notice that the forming of the above three inverses is easily accomplished since all three matrices are of fairly small dimension.)

A. Useful Properties of the Comparison Test Statistic

The *first useful property* to note is that the effect of time variation is actually suppressed in the forming of the $l_{ij}(t)$ as can be seen by a closer examination of the underlying structure, as now presented. Confining attention to only $l_{12}(t)$ (since identical arguments hold for the other l_{ij}), consider the following linear transformation[3] from the x to y domain:

$$y = [\bar{P}_{12}(t)]^{-1/2} [x_1(t) - x_2(t)]. \tag{A13}$$

Note that

$$E[y] = [\bar{P}_{12}(t)]^{-1/2} E[x_1(t) - x_2(t)] = 0 \tag{A14}$$

and

$$E[yy^{\mathrm{T}}] = \bar{P}_{12}(t)]^{-1/2} E\{[x_1(t) - x_2(t)] \times [x_1(t) - x_2(t)]^{\mathrm{T}}\} [\bar{P}_{12}(t)]^{-1/2} = [\bar{P}_{12}(t)]^{-1/2} \bar{P}_{12}(t) [\bar{P}_{12}(t)]^{-1/2} = I_{p \times p} \tag{A15}$$

and that the linear transformation of (A13) on the Gaussianly distributed $[x_1(t) - x_2(t)]$ guarantees that y is also Gaussianly distributed (and, as indicated from (A14) and (A15), of zero mean, and unity variance, and that the individual components of y are uncorrelated and thus *independent* since y is Gaussian). *Thus the statistics of $y(t)$ are time invariant* (which is why the time t was suppressed for y in (A13)). Therefore the measure

$$l_{12}(t) = y^{\mathrm{T}} y = y_1^2 + y_2^2 + \cdots + y_p^2 \tag{A16}$$

[3] Although a square root $[\bar{P}_{12}(t)]^{-1/2}$ is indicated above, no such computational requirement explicitly exists for real-time mechanization.

has statistics that are time invariant (and similarly for the other l_{ij}).

Eqs. A13 to A15 are actually an abbreviated form (that encompasses two different cases) that is merely offered here to facilitate transfer of the main idea. The two cases currently subsumed in Eqs. A13 to A15 (with a slight abuse of notation) that should rightly be treated separately to be totally rigorous are for raw data voting and for filtered data voting, respectively, and a distinction should be made as to whether the total expectation is being used or just the conditional expectation, given the measurements. In the case of having both raw data and filtered estimates available, voting can take place on

$$[(w_1 - \hat{x}_1) - (w_2 - \hat{x}_2)] \equiv (x_1 - \hat{x}_1) - (x_2 - \hat{x}_2)$$

by virtue of Eq. A4, and the result corresponding to Eq. A14 being zero is the same with either total or the above mentioned conditional expectations being utilized throughout since for Kalman filtering, the conditional covariance can be calculated a priori and it is identical to the covariance under total expectation [233].

A second useful property of the $l_{ij}(t)$ is that *they are central chi-square distributed with p-degrees of freedom.*[4] This property follows directly from the definition of central chi-square [84, p. 109], where the number of terms in the sum fixes the number of degrees of freedom. The requirements for $l_{12}(t)$ to be chi-squared of p degrees of freedom (χ_p^2) is satisfied by examining (A13) to (A15) (and the same conclusions follow for the other l_{ij} by identical arguments). Constant decision "threshold of goodness" γ_{ij}, can be obtained from statistical tables for central chi-square (e.g., [84, pp. 601, 602, Table 2]) such that the value of $l_{ij}(t_k)$, as calculated from the measurements should be

$$l_{ij}(t_k) \le \gamma_{ij} \tag{A17}$$

with probability of a set value α (say, 0.95). If any $l_{ij}(t_k)$ exceeds this constant value, then it is an indication that

[4]Conversely, for a known discrepancy between the means of the two estimates, the distribution of l_{ij} is noncentral chi square. This aspect can be utilized in quantifying detection probabilities.

(a) Case 1: 3 subsystems available.

Component Subsystem	1	2	3
Test Statistics Explicitly Evaluated		l_{12}	l_{13}
			l_{23}

Note: $l_{31} = l_{13}$, $l_{21} = l_{12}$, $l_{32} = l_{23}$

(b) Case 2: 4 subsystems available.

Component Subsystem	1	2	3	4
Test Statistics Explicitly Evaluated		l_{12}	l_{13}	l_{14}
			l_{23}	l_{24}
				l_{34}

(c) Case 3: 5 subsystems available.

Component Subsystem	1	2	3	4	5
Test Statistics Explicitly Evaluated		l_{12}	l_{13}	l_{14}	l_{15}
			l_{23}	l_{24}	l_{25}
				l_{34}	l_{35}
					l_{45}

(d) Case N: N subsystems available.

Component Subsystem	1	2	3	4	5 . . . N
Test Statistics Explicitly Evaluated		l_{12}	l_{13}	l_{14}	l_{15} . . . l_{1N}
			l_{23}	l_{24}	l_{25} . . . l_{2N}
				l_{34}	l_{35} . . . l_{3N}
					⋮
					$l_{N-2,N-1}$ $l_{N-2,N}$
					$l_{N-1,N}$

Note: Only $N(N-1)/2$ explicit evaluations required (as upper triangle of $(N-1) \times (N-1)$ square) since lower triangle is identical by symmetry of data compatibility testing: $l_{i,j} = l_{j,i}$

Fig. 10. Generalization of requisite test statistic calculations needed for cross-comparing subsystems of comparable measurements.

TABLE IV
Conceptual Basis for the Voter Tallying Algorithm

Reporting Subsystems \ Subsystem Under Review	1	2	. . .	N
1	0	VOTE(2,1)		VOTE(N,1)
2	VOTE(1,2)	0		VOTE(N,2)
.				
.				
.				
N	VOTE(1,N)	VOTE(2,N)		0
TALLIES	TALLY(1)	TALLY(2)	. . .	TALLY(N)

$$\text{VOTE}(i,j) = \begin{cases} 0 \\ 1 \end{cases}$$

$$0 \leq \text{TALLY}(i) \leq N-1$$

$$\text{TALLY}(i) = \sum_{j \neq i}^{N} \text{VOTE}(i,j)$$

something is wrong or that the two subsystems i and j have incompatible indications.

For the case of a known but calibrated random bias being present in Eq. A14 in the nominal unfailed situation so that it is no longer zero. This can still be handled using noncentral chi-square where the noncentrality parameter appropriately accounts for this.

A voting scheme that is reasonably easy to understand, interpret, and justify is offered in Section III using the terminology and definitions of this appendix.

These measures of intersubsystem sensor data consistency or discrepancy (depending, respectively, on whether the test statistic is below or above the appropriate constant decision threshold level) serves as the basis of the voter/monitor methodology that was described in Section III.

B. Voter/Monitoring for Both Raw Data and Filtered Data

The test statistics for pair-wise consistency checking, as described in the preceding section, can be applied as long as the requisite covariances are supplied along with the Gaussianly distributed estimates. The total number of explicit pair-wise cross-comparisons that need be made for an exhaustive examination are depicted in Figs. 10(a), (b), (c) for three case studies involving 3, 4, and 5 component subsystems, respectively. The number and type of cross-comparisons that need be performed for an exhaustive examination in the general case involving N sensor subsystems is depicted in Fig. 10(d) and requires only $N(N-1)/2$ explicit evaluations. Such an evaluation appears to not constitute an unreasonable computational burden in most SMN applications since usually fewer than 8 subsystems would need be checked in this manner in a worst case loading yielding only $8 \cdot 7/2 = 28$ evaluations that need be made. (A more likely nominal load of 3 or 4 subsystems being present would only require 3 to 5 evaluations!)

The flowchart of an algorithm for performing the pairwise subsystem comparisons for SMN is presented in Fig. 3. The SMN subsystem sensor data is to be in the format described in Table II. This algorithm (and the subsequent algorithm described in Fig. 4) is as applicable to raw data testing as it is to filtered data testing.

A conceptual tallying procedure is depicted in Table IV that served as the basis of the vote tallying algorithm offered in Fig. 4. The entire voter/monitoring procedure is covered in the flowcharts of Figs. 3 and 4 with the philosophy as described in detail in Section III that justifies its tailoring to the SMN application. A few additional design issues are discussed below.

It may be desirable that cross-comparison voting not be carried out separately for the horizontal and vertical components of position even though more uncertainty is anticipated to be in the vertical components as usually occurs for most navigation sensors. The reason that such fine resolution was avoided here is that if vertical and horizontal channels were voted separately, differing voting outcomes do not offer the logical follow-on of mixing and matching in real time; however, using all three components in a test, when appropriate, allows all cross-correlations to be adequately accounted for as a test of greater significance. Similar remarks can be made about the degree of isolation refinement required in real-time for on-line SMN operation.

In conclusion, for those sensor subsystems that do not offer on-line computation of the associated covariance that is necessary to have before it can participate in the voter/monitor methodology described here, other means of inputting the necessary covariances are routinely available at both an aggregate system level as illustrated in Table V and at a detailed level as in Tables VI, VII, and VIII. Rules-of-thumb have also been summarized as in Table IX as one way to decide whether a secondary subsystem should even be participating in the voter/monitoring methodology or be more properly classified as not available yet until sufficiently within range of its broadcast station.

TABLE V
Composite Performance Comparison of Modern Radio Navigation Systems [100]

	Vortac (VOR + TACAN)	Doppler	JTIDS RELNAV	GPS
Volumetric coverage	Line-of-sight (100 NM)	Unlimited	Line-of-sight	Global to ground
Signal reliability	High VHF, L-band	Moderate (13 GHz)	High (L-band) 960-1215 MHz	High (1227, 1575 MHz)
Data Content	Relative Rho/ Theta 2D Pos.	2D Position 3D Velocity	Relative 2D Pos., 3D Pos. (1), 2D Vel., Absol 2D Pos. (2), time (3)	Absolute 3D Pos., 3D Vel, time
Accuracy	$\pm$1.65, 1σ $\pm$1NM $\pm$600 ft 1σ(DME)	0.1-0.25% 1σ (Velocity) 0.5-1σCEP (Position)	Classified	25 ft Hor. Pos. (6) 35 ft Vert. Pos. (6) 0.1 Knot Vel. (6)
Application Versatility	Air, short distance approach	Air, long dist. dead-reckoning	Air, surface med dist. Coll. avoidance, weap. delivery	Air, surface, space, long dis. terminal, weap. delivery
User equipment cost	Low	Moderate	Moderate (7)	Moderate to high (8)

(1) 3D data available for favorable geometry of sources only.
(2) If position references are available.
(3) Under optimum conditions, 100 ft CEP considered likely.
(4) Dual frequency receiver, exclusive of own velocity error effect.
(5) Single frequency receiver, exclusive of own velocity error effect.
(6) Predicted for highest performance receiver and full 21-satellite configuration.
(7) RELNAV function is software addition only to basic communication terminal.
(8) Different performance quality level equipment being developed.

TABLE IX
Sensor Range and Accuracy [100]

		Error $\pm$ degrees or $\pm$ feet			
SYSTEM	RANGE, n mi	PROPAGATION	SITE*	INSTRUMENT	ACCEPTED SYSTEM
VOR	200α	Negligible	3°	1°	3.5°
Doppler VOR	200α	Negligible	0.5°	1°	1.5°
DME	200	Negligible	None	200 ft. 2%†	3000 ft or 3%
Tacan:					
Range	200	Negligible	None	200 to 2000 ft	2000 ft
Bearing	200	Negligible	2	0.5	2

* Typical

α Line of sight

† Depending on price

TABLE VI
Gimbaled Platform INS Error Model [103]

Description	RMS VALUE
TUNED ROTOR GYROS	
Fixed Drift Rate Biases	0.01 deg/hr
Scale-Factor Errors	400 ppm
G-Sensitive Drift Rate Biases	
Mass Unbalance	0.02 deg/hr/g
Quadrature	0.02 deg/hr/g
Heading-Sensitive Drift Rate Biases	
Level Axes	0.007 deg/hr
Vertical Axes	0.025 deg/hr
Input Axes Misalignment Angles	
Level Axes	50 arc sec
Vertical Axes	150 arc sec
ACCELEROMETERS	
Fixed Biases	50 μg
Scale-Factor Errors	300 ppm
Input Axes Misalignment Angles	80 arc sec
Vertical Deflections*	8.3 arc sec

*Correlation distance = 25 nm

TABLE VII
Strapdown INS Error Model [103]

Description	RMS VALUE
RING LASER GYROS	
Fixed Drift Rate Bias	0.008 deg/hr
Scale-Factor Error	2 ppm
Wideband Noise	0.002 deg/ hr
Scale-Factor Linear Asymmetry	(0.2 deg/hr)/(rad/sec)^2
Scale-Factor Omega-Squared Asymmetry	
Input Axies Misalignment Angles	5 arc sec
ACCELEROMETERS	
Fixed Biases	40 μg
Scale-Factor Errors	60 ppm
Scale-Factor Linear Asymmetry	40 ppm
Input Axis Mis-alignment Angles	
Yaw	7 arc sec
Pitch and Roll	10 arc sec
Random Noise*	3 μg
Vertical Deflections+	8.3 arc sec

* Correlation time = 60 sec

+ Correlation distance = 25 nm

TABLE VIII
NAVAID Error Models [103]

Description	RMS VALUE
DOPPLER RADAR	
Misalignment Angles	3.4 arc min
Scale-Factor Errors	0.1 %
Pitch Calibration Error	17 arc sec
Wideband Error*	
Along-Track	0.0084 $\sqrt{v_g}$ ft/sec
Cross-Track	0.0143 $\sqrt{v_g}$ ft/sec
Sea Surface Current+	1.3 ft/sec
GPS	
Wideband Position Errors	14 ft
Wideband Velocity Errors	0.06 ft/sec

* Discrete measurement errors (60 sec averaging), v_g is ground speed in ft/sec.

+ Over-water segment only, correlation time = 1hr

REFERENCES

[1] Newbold, P.M., and Ho, Y.C. (1968)
Detection of changes in the characteristics of a Gauss–Markov process.
IEEE Transactions on Aerospace and Electronic Systems, AES-4, 5 (Sept. 1968), 707–718.

[2] Wald, A. (1947)
Sequential Analysis.
New York: Wiley, 1947.

[3] Prabhu, K.P. (1970)
On the detection of a sudden change in system parameters.
IEEE Transactions on Information Theory (Corresp.) (July 1970), 497–500.

[4] Phatak, A.V. (1971)
Comments on "On the detection of a sudden change in system parameters."
IEEE Transactions on Information Theory (Corresp.) (May 1971), 349.

[5] Prabhu, K.P.S. (1971)
Author's reply.
IEEE Transactions on Information Theory (Corresp.) (May 1971), 249.

[6] Chien, T.T., and Adams, M.B. (1976)
A sequential failure detection technique and its application.
IEEE Transactions on Automatic Control, AC-21, 5 (Oct. 1976), 750–757.

[7] Deckert, J.C., Desai, M.N., Deyst, J.J., and Willsky, A.S. (1977)
F-8 DFBW sensor failure identification using analytic redundancy.
IEEE Transactions on Automatic Cotrol, AC-22, 5 (Oct. 1977), 795–803.

[8] McAulay, R.J., and Denlinger, E. (1973)
A decision-directed adaptive tracker.
IEEE Transactions on Aerospace and Electronic Systems, AES-9, 2 (Mar. 1973), 229–236.

[9] Stuller, J.A. (1975)
Generalized likelihood signal resolution.
IEEE Transactions on Information Theory, IT-21, 3 (May 1975).

[10] Willsky, A.S., and Jones, H.L. (1976)
A generalized likelihood ratio approach to detection and estimation of jumps in linear systems.
IEEE Transactions on Automatic Control, AC-21, 1 (Feb. 1976), 108–112.

[11] Willsky, A.S., and Jones, H.L. (1974)
A Generalized Likelihood Ratio Approach to State Estimation in Linear Systems Subject to Abrupt Changes.
Report ESL-P-538, M.I.T. Electronic Systems Laboratory, Cambridge, Mass., 1974.

[12] Willsky, A.S. (1976)
A survey of design methods for failure detection in dynamic systems.
Automatica, 12 (Nov. 1976), 601–611.

[13] Segan, J., and Sanderson, A.C. (1980)
Detecting change in a time-series.
IEEE Transactions on Information Theory, IT-26, 2 (Mar. 1980), 249–255.

[14] Kerr, T.H. (1974)
A two ellipsoid overlap test for real-time failure detection and isolation by confidence regions.
Proceedings of the IEEE Conference on Decision and Control (1974), 735–742.

[15] Kerr, T.H. (1977)
Real-time failure detection: A nonlinear optimization problem that yields a two-ellipsoid overlap test.
Journal of Optimization Theory and Applications, 22, 4 (Aug. 1977), 509–535.

[16] Kerr, T.H. (1976)
Failure detection aids for human operator decisions in a precision inertial navigation system complex.
Proceedings of Symposium on Applications of Decision Theory to Problems of Diagnosis and Repair (American Statistical Association, Fairborn, Ohio, June 2–3, 1976), 98–127.

[17] Kerr, T.H. (1980)
Statistical analysis of a two ellipsoid overlap test for real-time failure detection.
IEEE Transactions on Automatic Control, AC-25, 4 (Aug. 1980), 762–773.

[18] Kerr, T.H. (1982)
False alarm and correct detection probabilities over a time interval for restricted classes of failure detection algorithms.
IEEE Transactions on Information Theory, IT-20, 4 (July 1982), 619–631.

[19] Kerr, T.H. (1974)
Poseidon improvement studies: Real Time Failure Detection in the SINS/ESGM.
Technical Report TR-418-2, TASC, June 1974 (Confidential).

[20] Kerr, T.H. (1975)
Failure Detection in the SINS/ESGM System.
Technical Report TR-528-3-1, The Analytic Sciences Corporation, Reading, Mass., July 1975 (Confidential).

[21] Kerr, T.H. (1976)
Improving ESGM Failure Detection in the SINS/ESGM System (U).
Technical Report TR-678-3-1, The Analytic Sciences Corporation, Reading, Mass., Oct. 1976 (Confidential).

[22] Kerr, T.H. (1977)
TASC Refinements of SINS/ESGM Failure Detection Procedures.
TASC slide presentation SP-889-3-1, for Sperry Systems Management, Feb. 1977.

[23] Willsky, A.S., Chow, E.Y., et al. (1980)
Dynamic model-based techniques for the detection of incidents on freeways.
IEEE Transactions on Automatic Control, AC-25, 3 (June 1980), 347–361.

[24] Duiven, E.M., Medler, C.L., and Kapser, J.F. (1981)
Use of filtering and smoothing algorithms in the analysis of missile system test data, and validation of filter/smoother models.
In C.T. Leondes (Ed.), *Advances in the Theory and Technology of Applications of Nonlinear Filters and Kalman Filters*. NATO Advisory Group for Aerospace Research and Development. Leiden, The Netherlands: AGARDograph Noordhoff International Publishing, 1981.

[25] Caglayan, A.K., and Lancraft, R.E. (1982)
An Aircraft Sensor Fault Tolerant System.
NASA Contractor Report 165976 by Bolt Beranek and Newman, Inc., Cambridge, Mass., Apr. 1982.

[26] Caglayan, A.K. (1980)
Necessary and sufficient conditions for detectability of jumps in linear systems.
IEEE Transactions on Automatic Control, AC-25, 4 (Aug. 1980), 833–834.

[27] Chow, E.Y. (1976)
Analytical studies of the generalized likelihood ratio technique for failure detection.
M.S. Thesis, Massachusetts Institute of Technology, Cambridge, Mass., Feb. 1976.

[28] Harrison, J.V., Daly, K.C., and Gai, E. (1980)
Reliability and accuracy prediction for a redundant strapdown navigator.
In *Proceedings of the AIAA Guidance and Control Conference* (Danvers, Mass., Aug. 11–13, 1980), 403–413.

[29] Chan, Y.K. and Edsinger, R.W. (1981)
A correlated random number generator and its use to estimate false alarm rates of airplane sensor failure detection algorithms.
IEEE Transactions on Automatic Control, AC-256, 3 (June 1981), 676–680.

[30] Shooman, M.L. (1968)
Probabilistic Reliability: An Engineering Approach.
New York: McGraw-Hill, 1968.

[31] Willsky, A.S., and Jones, H.L. (1974)
A generalized likelihood ratio approach to state estimation in linear systems subject to abrupt changes.
Proceedings of the IEEE Conference on Decision and Control (Phoenix, Ariz., Nov. 1974).

[32] Jones, H.L. (1973)
Failure detection in linear systems.
Ph.D. Dissertation, Aeronautics and Astronautics Department, Massachusetts Institute of Technology, Cambridge, Mass., Draper Report T-608, Aug. 1973.

[33] Friedland, B. (1979)
Maximum-likelihood estimation of a process with random transitions (failures).
IEEE Transactions on Automatic Control, AC-24, 6 (Dec. 1979), 932–937.

[34] Friedland, B. (1981)
Multidimensional maximum-likelihood failure detection and estimation.
IEEE Transactions on Automatic Control, AC-26, 2 (Apr. 1981), 567–570.

[35] Friedland, B., and Grabousky, S. (1982)
Estimating sudden changes in biases in linear dynamic systems.
IEEE Transactions on Automatic Control, AC-27, 1 (Feb. 1982), 237–240.

[36] Friedland, B. (1982)
Maximum likelihood failure detection of aircraft flight control sensors.
Journal of Guidance, Control, and Dynamics, 5, 5 (Sept.–Oct. 1982), 498–503.

[37] Beard, R.V. (1971)
Failure accommodation in linear systems through self-reorganization.
Ph.D. Dissertation MTV-71-1, Aeronautics and Astronautics Department, Massachusetts Institute of Technology, Cambridge, Feb. 1971.

[38] Gai, E., Harrison, J.V., and Daly, K.C. (1978)
Failure detection and isolation performance of two redundant sensor configurations.
In *Proceedings of Position Location and Navigation Symposium (PLANS)* (San Diego, Calif., 1978), 122–131.

[39] Gai, E.G., and Adams, M.B. (1976)
Determination of failure thresholds in hybrid navigation.
IEEE Transactions on Aerospace and Electronic Systems, AES-12, 6 (Nov. 1976), 744–755.

[40] Motyka, P.R., and Bell, J.W. (1980)
Failure detection and isolating for tactical aircraft using the generalized likelihood test.
In *Proceedings of the IEEE 1980 National Aerospace and Electronics Conference (NAECON)* (Dayton, Ohio, May 1980), 20–22.

[41] Hertel, J.E., and Clark, R.N. (1982)
Instrument failure detection in partially observable systems.
IEEE Transactions on Aerospace and Electronics Systems, AES-18, 3 (May 1982), 310–317.

[42] Chang, C.B., and Dunn, K.P. (1979)
On the GLR detection and estimation of unexpected inputs in linear discrete systems.
IEEE Transactions on Automatic Control, AC-24, 3 (June 1979), 499–500.

[43] Tait, K.S., Toohey, E.F., and Vahlberg, C.J. (1977)
Improved Guidance Study Results: 2nd Interim Report.
Technical Report TR-836-2, The Analytic Sciences Corporation, Nov. 1977 (Confidential).

[44] Talwar, P.P., and Gentle, J.E. (1981)
Detecting a scale shift in a random sequence at an unknown time point.
Applied Statistics, 30, 3 (1981), 301–304.

[45] Hibey, J.L. (1981)
Performance analysis for differential systems incurring abrupt random changes.
IEEE Transactions on Automatic Control, AC-26, 2 (Apr. 1981), 553–554.

[46] Fiorina, M., and Maffezzoni, C. (1979)
A direct approach to jump detection in linear time-invariant systems with application to power system perturbation.
IEEE Transactions on Automatic Control, AC-24, 3 (June 1979), 428–433.

[47] Stalford, H. (1982)
A computationally efficient GLR algorithm for detecting and estimating state jumps in linear systems.
In *Proceedings of the IFAC Symposium on Identification and System Parameter Estimation* (June 7–11, 1982), 827–832.

[48] Kerr, T.H. (1983)
Examining the controversy over the acceptability of SPRT and GLR in some applications.
In *Proceedings of the American Control Conference* (San Francisco, Calif., June 22–24, 1983).

[49] Walker, B. (1983)
Recent developments in fault diagnosis and accommodation.
Presented at the AIAA Guidance and Control Conference, Aug. 1983.

[50] Potter, J.E., and Suman, M.C. (1977)
Thresholdless redundancy management with arrays of skewed instruments.
In Integrity in Electronic Flight Control Systems, NATO AGARDograph 224, Paper 15, Apr. 1977.

[51] Hall, S.R., Motyka, P., Gai, E., and Deyst, J.J. (1983)
In-flight parity vector compensation for FDI.
IEEE Transactions on Aerospace and Electronics Systems, AES-19, 5 (Sept. 1983), 668–675.

[52] Maybeck, P. (1976)
Failure Detection Through Functional Redundancy.
Technical Report AFFDL-TR-76-93, Air Force Flight Dynamics Laboratory, Final Report for Period January 1972 to July 1973, Sept. 1976.

[53] Kubbat, W.J. (1979)
Failure detection, isolation and indication in highly integrated digital guidance and control systems.
In Advances in Guidance and Control Systems Using Digital Techniques, NATO AGARDograph AGARD-CP-272, May 1979.

[54] Teaque, T.L., and Powers, G.J. (1981)
Diagnosis procedures from fault tree analysis.
In *Proceedings of the Automatic Control Conference* (Charlottesville, Va., June 17–19, 1981).

[55] Folkesson, K. (1980)
Computer-based in-flight monitoring.
In Fault Tolerance Design and Redundancy Management Techniques, NATO AGARD Lecture AGARD-LS-109, Oct. 1980.

[56] Cunningham, T.B. (1980)
Failure management techniques for high survivability.
In Fault Tolerance Design and Redundancy Management Techniques, NATO AGARD Lecture Series AGARD-LS-109, Oct. 1980.

[57] Noble, W.B. (1983)
Fault isolation methodology for the L-1011 digital avionic flight control system.
Journal of Guidance Control, and Dynamics, 6, 2 (Mar.–Apr. 1983), 72–76.

[58] Li, D.S., Pan, Q.R., and Singh, M.G. (1982)
A Novel Approach to the Identification of Failed Sensors in Dynamical Systems.
Control Systems Center Report 532, University of Manchester Institute of Science and Technology, Manchester, England, Jan. 1982.

[59] Motyka, P. (1981)
Fault detection for two physically separate communicating inertial measurement units.
In *Proceedings of the Automatic Control Conference* (Charlottesville, Va., June 17–19, 1981).

[60] Szalai, K.J., Larson, R.R., and Glover, R.D. (1980)
Flight experience with flight control redundancy management.
In Fault Tolerance Design and Redundancy Management Techniques, NATO AGARD Lecture Series AGARD-LS-109, Oct. 1980.

[61] Korn, J., and Gully, S.W. (1982)
Application of the generalized likelihood ratio algorithm to maneuver detection and estimation.
In *Proceedings of the 1982 American Control Conference* (Arlington, Va., June 14–16, 1982).

[62] Speyer, J.L., and White, J.E. (1984)
The Shiryayev sequential probability ratio test for redundancy management.
Journal of Guidance, Control, and Dynamics, 7, 5 (Sept./Oct. 1984), 588–595.

[63] Luppold, R.H., Gai, E., and Walker, B.K. (1984)
Effects of redundancy management of reliability modeling.
In *Proceedings of the 1984 American Control Conference* (San Diego, Calif., June 6–8, 1984), 1763–1779.

[64] Gregson, F. (1972)
Fault detection possibility in a system employing Kalman filtering.
In Inertial Navigation Components and Systems, NATO AGARD, AGARD-CF-114, Oct. 2–5, 1972.

[65] Savage, P.G. (1982)
Strapdown inertial navigation overview, presentation and handouts for short course.
Presented at the 1982 Position, Location, and Navigation Symposium (PLANS), Dec. 6, 1982.

[66] Schwartz, H.M. (1983–1984)
Sensitivity analysis of an integrated NAVSTAR GPS/INS navigation system to component failure.
Navigation: Journal of the Institute of Navigation, 30, 4 (Winter issue, 1983–1984), 325–337.

[67] Yanney, R.M., and Endo, N.M. (1984)
Built-in-test for military computers.
In *Mini/Micro Northeast, Professional Program Session Record*, No. 24, paper 1 (May 17, 1984), 1–4.

[68] Madiwale, A.N., and Friedland, B. (1983)
Comparison of innovations-based failure detection and correction methods.
In *Proceedings of the American Control Conference* (San Francisco, Calif., June 22–24, 1983).

[69] Caglayan, A., and Lancraft, R.E. (1983)
Reinitialization issues in fault tolerant sytems.
In *Proceedings of the American Control Conference* (San Francisco, Calif., June 22–24, 1983).

[70] Tugnait, J.K. (1983)
Detection and identification of abrupt changes in linear systems.
In *Proceedings of the American Control Conference* (San Francisco, Calif., June 22–29, 1983).

[71] Corsini, G., Messe, E.D., Marchetti, G., and Verrazzani, L. (1984)
Comments on "On a direct method of analysis and synthesis of the SPRT."
IEEE Transactions on Information Theory, IT-30, 3 (May 1984), 573–574.

[72] Lou, Z.I.-C., and Willsky, A.S. (1983)
Failure detection with uncertain models.
In *Proceedings of the American Control Conference* (San Francisco, Calif., June 22–29, 1983).

[73] Leininger, G.G. (1981)
Model degradation of sensor failure detection.
In *Proceedings of 1981 Joint Automatic Control Conference* (Charlottesville, Va., June 17–19, 1981).

[74] Chien, T.T. (1972)
An Adaptive Technique for a Redundant Sensor Navigation System.
Report T-560, Charles Stark Draper Lab., Feb. 1972.

[75] Liu, J.H.S., and Jones, H.L. (1977)
Linear manifold constrained GLR.
IEEE Transactions on Automatic Control, AC-22, 6 (Dec. 1977), 988–989.

[76] Walker, B.K. (1981)
Performance evaluation of systems that include fault diagnostics.
Presented at the 1981 Joint Automatic Control Conference, Charlottesville, Va., June 1981.

[77] Daly, K.C., Gai, E., and Harrison, J.V. (1979)
Generalized likelihood test for FDI in redundant sensor configurations.
Journal of Guidance and Control, 2, 1 (Jan.–Feb. 1979), 9–17.

[78] Broen, R.B. (1974)
A nonlinear voter-estimator for redundant systems.
In *Proceedings of the IEEE Conference on Decision and Control* (Dec. 1974), 743–748.

[79] Gai, E., Harrison, J.V., and Deyst, J.J., et al. (1981)
Availability analysis of a redundant data collection and display system for nuclear power plants.
In *Proceedings of the Automatic Control Conference* (Charlottesville, Va., June 17–19, 1981).

[80] Mehra, R.K., and Peschon, J. (1971)
An innovations approach to fault detection and diagnosis in dynamic systems.
Automatica, 7, 5 (1971).

[81] Boozer, D.D., and McDaniel, W.L. (1972)
On innovation sequence testing of the Kalman filter.
IEEE Transctions on Automatic Control, AC-17, 1 (1972).

[82] Martin, W.C., and Stubberud, A.R. (1974)
An additional requirement for innovations testings in system identification.
IEEE Transactions on Automatic Control, AC-19, 5 (1974).

[83] Maybeck, P.S. (1974)
Filter Design for a TACAN-aided Baro-Inertial System with ILS Smoothing Capability.
Report AFFDL-TM-74-52, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, Jan. 1974.

[84] Bowker, A.H., and Lieberman, G.J. (1972)
Engineering Statistics.
Englewood Cliffs, N.J.: Prentice-Hall, 1972.

[85] Proposed Engineering Development Model for INS/GPS Integration. (1983)
Litton Aero Product Division Proposal 600347, submitted to Naval Air System Command, May 30, 1983.

[86] Speyer, J.L. (1979)
Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem.
IEEE Transactions on Automatic Control, AC-24, 2 (Apr. 1979).

[87] Chang, T.S. (1980)
Comments on "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control."
IEEE Transactions on Automatic Control, AC-25, 3 (June 1980).

[88] Willsky, A., Bello, M., Castanon, B.C., Levy, G., and Verghese, G. (1982)
Combining and updating of local estimates and regional maps along sets of one-dimensional tracks).
IEEE Transactions on Automatic Control, AC-27, 4 (Aug. 1982).

[89] Levy, B.U., Castanon, D.A., Verghese, G.C., and Willsky, A.S. (1983)
A scattering framework for decentralized estimation problems.
Automatica, 19, 4 (1983), 373–387.

[90] Bar-Itzhack, I.Y. (1980)
A novel method for data compression in recursive INS estimation.
Journal of Guidance and Control, 3 (May–June 1980), 245–250.

[91] Medan, Y., and Bar-Itzhack, I.Y. (1981)
Error and sensitivity analysis of a new data compression technique in estimation.
Journal of Guidance and Control, 4, 5 (Sept.–Oct. 1981), 510–517.

[92] Medan, Y., and Bar-Itzhack, I.Y. (1982)
Optimal batch recursive data compression.
IEEE Transactions on Automatic Control, AC-27, 1 (Feb. 1982), 169–176.

[93] Medan, Y., and Bar-Itzhack, I.Y. (1980)
Error Sensitivity Analysis Scheme of a New Data Compression Technique in Estimation.
Report TAE 401, Department of Aeronautical Engineering, Technion-Israel Institute of Technology, May 1980.

[94] Bar-Itzhack, I.Y., and Medan, Y. (1983)
Efficient square-root algorithm for measurement update in Kalman filtering.
Journal of Guidance, Control, and Dynamics, 6, 3 (May–June 1983), 129–134.

[95] Bar-Itzhack, I.Y. (1982)
Minimal order time sharing filters for INS in-flight alignment.
Journal of Guidance and Control, 5, 44 (July/Aug. 1982), 396–402.

[96] Kurien, T. (1983)
A partitioned data compression algorithm.
Ph.D. Dissertation, University of Connecticut, 1983.

[97] Kerr, T.H., and Chin, L. (1981)
The theory and techniques of discrete-time decentralized filters.
In C.T. Leondes (Ed.), *Advances in the Techniques and Technology of the Application of Nonlinear Filters and Kalman Filters*, NATO Advisory Group for Aerospace Research and Development AGARDograph.
Leiden, The Netherlands: Noordhoff International Publishing, 1981.

[98] Mossman, D.C., Higgins, W.T., Bochem, J.H., and Caldwell, D.G. (1984)
Suboptimal filtering for aided GPS navigation.
In *Proceedings of the ION Meeting* (Cambridge, Mass., June 25–28, 1984).

[99] Karatsinides, S.P., and Bush, R.L. (1984)
Navigation processing of the flight management computer system for the Boeing 737-300.
In *Proceedings of the ION Meeting* (Cambridge, Mass., June 26–28, 1984).

[100] Kayton, M., and Fried, W.R. (1969)
Avionics Navigation Systems.
New York: Wiley, 1969.

[101] Helfrick, A. (1984)
Modern Aviation Electronics.
Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[102] Gallo, M. (1982)
Development of an Air Force RLG strapdown standard navigator.
In *Proceedings of 1982 IEEE Position, Location, and Navigation Symposium* (Atlantic City, N.J., Dec. 6–9, 1982).

[103] Griffiths, B.E., Geyer, E.M., and Bernings, S.L. (1984)
Interfacing Kalman filters with standard INS.
In *Proceedings of IEEE NAECON* (Dayton, Ohio, May 21–25, 1984).

[104] Castanon, D.A., and Teneketzis, D. (1984)
Distributed estimation algorithms for nonlinear systems.
IEEE Transactions on Automatic Control, 1984.

[105] Kerr, T.H. (1983)
Phase III GPS Integration Options Volume 1: GPS User Equipment Characteristics.
Report IR-MA-177, Intermetrics, Inc., Cambridge, Mass., Jan. 26, 1983.

[106] Maybeck, P.S. (1979)
Stochastic Models Estimation and Control, vol. 1.
New York: Academic Press, 1979.

[107] Klinger, A. (1968)
Prior information and bias in sequential estimation.
IEEE Transactions on Automatic Control, AC-13, 1 (Feb. 1968), 102–103.

[108] Chong, C.Y. (1979)
Hierarchical estimation.
In *Proceedings of 2nd MIT/ONE Workshop on Distributed Communications and Decision Problems Motivated by Naval* C^3 *Problems*, vol. 1 (Naval Postgraduate School, Monterey, Calif., 1979), 205–220.

[109] Rome, H.J., Reilly, R.A., et al. (1980)
Synergistic benefits of fully integrating JTIDS and GPS.
In *Proceedings of NAECON* (May 1980), 780–791; update in 1982 *PLANS*.

[110] Copps, E.M., Geier, G.J., Fidler, W.C., and Grundy, P.A. (1980)
Optimal processing of GPS signals.
Navigation: Journal of the Institute of Naviation, 27, 3 (Fall 1980), 171–182.

[111] Hitt, E.F., Webb, J., Lucius, C., Bridgman, M.S., and Eldredge, D. (1983)
Handbook—Volume 1: Validation of Digital Systems in Avionics and Flight Control Applications.
Document DOT/FAA/CT-82/115, by Battelle Columbus Laboratories, Columbus, Ohio, and FAA Technical Center, Atlantic City Airport, N.J., for FAA, July 1983 (still in revision).

[112] Bierman, G.J. (1984)
Combining Estimates Obtained from Independent Data Sets.
Interoffice Memorandum 335.4-405, Jet Propulstion Laboratory, Pasadena, Calif., Mar. 20, 1984.

[113] Bierman, G.J. (1984)
Combining Estimates from Independent Data Sets Part II: Implementation.
Interoffice Memorandum 335.4-409, Jet Propulsion Laboratory, Pasadena, Calif., Apr. 23, 1984.

[114] Wax, M. (1983)
Position location from sensors with position uncertainty.
IEEE Transactions on Aerospace and Electronic Systems, AES-19, 5 (Sept. 1983), 658–662.

[115] Carlson, N.A., Kerr, T.H., and Sacks, J.E. (1984)
Integrated Navigation Concept Study.
Technical Report IR-MA-321, Intermetrics Inc., Cambridge, Mass., June 15, 1984.

[116] Intermetrics (1980)
Description of Current Relative Navigation (RelNav) Algorithms Final Report.
Report IR-434, Intermetrics, Inc., Cambridge, Mass., Mar. 31, 1980.

[117] Kung, H.T. (1982)
Why Systolic architectures?
IEEE Transactions on Computers (Jan. 1982), 37–45.

[118] Kung, S.Y., Whitehouse, H.J., and Kailath, T., (Eds.) (1985)
VLSI and Modern Signal Processing.
Englewood Cliffs, N.J.: Prentice-Hall, 1985.

[119] Hitt, E.F., and Webb, J.J. (1984)
A fault-tolerant software strategy for digital systems.
In *Proceedings of Digital Avionics System Conference* (Baltimore, Md., Dec. 3–6, 1984), 211–216.

[120] Mendel, J.M. (1971)
Computational requirements for a discrete Kalman filter.
IEEE Transactions on Automatic Control, AC-16, 6 (1971), 748–759.

[121] Gura, I.A., and Bierman, A.B. (1971)
On computational efficiency of linear filtering algorithms.
Automatica, 7 (1971), 299–314.

[122] Kriegsman, B.A., and Stonestreet, W.M. (1978)
A navigation filter for an integrated GPS/JTIDS/INS system for a tactical aircraft.
In *Proceedings of the IEEE Position, Location, and Navigation Symposium (PLANS)* (San Diego, Calif., Nov. 6–9, 1978), 316–320.

[123] TRW (1980)
MFBARS Radio Architecture Study: Final Draft.
Technical Report, TRW, Redondo Beach, Calif., Mar. 1980.

[124] ITT Avionics (1980)
Modular Multifunction-Multiband Airborne Radio System (MFBARS) Preliminary Report.
Technical Report, ITT Avionics, Nutley, N.J., Jan. 15, 1980.

[125] Charles Stark Draper Lab. (1978)
GPS/JTIDS/INS Integration Study: Final Report, vol. II of IV.
Technical Report R-1151, The Charles Stark Draper Laboratory, Inc., Cambridge, Mass., June 1978.

[126] Travassos, R.H., and Andrews, A. (1983)
VLSI implementation of parallel Kalman filters.
In *Proceedings of the AIAA Guidance and Control Conference*, Advanced Avionics Session (San Diego, Calif., Aug. 1983), 643–649.

[127] Travassos, R.H. (1983)
Applications of Systolic Array Technology to Recursive Filtering.
In *Modern Signal Processing*.
Englewood Cliffs, N.J.: Prentice-Hall, 1983.

[128] Travassos, R.H. (1983)
Real-time implementation of systolic Kalman filters.
In *Proceedings of the Society of Photogammetry and Imaging Engineers (SPIE)* (San Diego, Calif., 1983), 1–8.

[129] Volder, J.E. (1959)
The CORDIC trigonometrics computing technique.
IRE Transactions on Electronic Computations (Sept. 1959), 330–334.

[130] Ahmed, H.M.
Convergence properties of CORDIC algorithms.
To appear.

[131] Van Leeuwen, J. (1983)
Distributed Computation.
In *Foundations of Computer Science IV* and *Distributed Systems: Part 1: Algorithms and Complexity*.
Amsterdam: Mathematical Centre Tracts 158, 1983.

[132] Valiant, L.G. (1983)
Parallel computation.
In J.W. DeBakker and J. Van Leeuwen (Eds.), *Foundations in Computer Science IV*.
Amsterdam: Mathematical Centre Tracts 158, 1983.

[133] Bandet, G.M. (1983)
Design and complexity of VLSI algorithms.
In J.W. DeBakker and J. Van Leeuwen (Eds.), *Foundations of Computer Science IV*.
Amsterdam: Mathematical Centre Tracts 158, 1983.

[134] Kramer, M.R., and Van Leeuwen, J. (1983)
Systolic computation and VLSI.
In J.W. DeBakker and J. Van Leeuwen (Eds.), *Foundations of Computer Science IV*.
Amsterdam: Mathematical Centre Tracts 158, 1983.

[135] Joglekar, A.N., and Powell, J.D. (1973)
Data compression in recursive estimation with applications to navigation systems.
In *Proceedings of AIAA Guidance and Control Conference* (Key Biscayne, Fla., Aug. 20–22, 1973).

[136] Widnall, W.S., and Gobbini, G. (1980)
Decentralized Relative Navigation and JTIDS/GPS/INS Integrated Navigation Systems, First Interim Technical Report.
Massachusetts Institute of Technology, Cambridge, June 10, 1980.

[137] Widnall, W.S., and Gobbini, G. (1980)
Decentralized Relative Navigation and JTIDS/GPS/INS Integrated-Navigation Systems, Second Interim Technical Report.
Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Nov. 15, 1980.

[138] Fitzgerald, R. (1982)
Simple tracking filters: Position and velocity measurements.
IEEE Transactions on Aerospace and Electronic Systems, AES-18, 5 (Sept. 1982), 531–538.

[139] Westbrook, E.A. (1979)
Integration of JTIDS and GPS in tactical aircraft.
Presented at IEEE EASCON, Oct./Nov. 1979.

[140] Widnall, W.S., Gobbini, G.F., and Kelley, J.F. (1982)
Decentralized Relative Navigation and JTIDS/GPS/INS Integrated Navigation Systems.
Report issued as AFWAL-TR-81-1264, Massachusetts Institute of Technology, Cambridge, Jan. 1982.

[141] Upadhyay, T.N., Kriegsman, B.A., and Cox, D.B. (1982)
Benefits of integrating GPS and inertial navigation systems.
Proceedings of Institute of Navigation Annual Meeting (Colorado Springs, Colo., June 1982).

[142] Perriot-Mathonna, D.M. (1984)
Improvements in the application of stochastic estimation algorithms—parameter jump detection.
IEEE Transactions on Automatic Control, AC-29, 11 (Nov. 1984), 962–969.

[143] Blank, R.W. (1983)
The NAVSTAR GPS full scale development user equipment family.
In *Proceedings of ION National Aerospace Meeting* (Arlington, Va., May 22–25, 1983), 65–72.

[144] Murphy, J.W., and Yakos, M.D. (1983)
Collins Avionics NAVSTAR receiver.
In *Proceedings of ION National Aerospace Meeting* (Arlington, Va., Mar.22–25, 1983), 107–116.

[145] Kao, M.H., and Eller, D.H. (1983)
Multiconfiguration Kalman filter design for high performance GPS navigation.
IEEE Transactions on Automatic Control, AC-28, 3 (Mar. 1983), 304–314.

[146] Papoulis, A. (1965)
Probability, Random Variables, and Stochastic Processes.
New York: McGraw-Hill, 1965.

[147] Falconer, D.D., and Ljung, L. (1978)
Application of fast Kalman estimation to adaptive equalization.
IEEE Transactions in Communication, COM-26 (Oct. 1978), 1439–1496.

[148] Porat, B. (1983)
Pure time updates for least squares lattice algorithms.
IEEE Transactions on Automatic Control, AC-28, 8 (Aug. 1983), 865–866.

[149] Schreiber, R., and Kuekes, P.J. (1983)
Systolic machines in digital signal processing.
In *Proceedings of the USC Workshop on VLSI and Modern Signal Processing*.
Englewood Cliffs, N.J.: Prentice-Hall, 1983.

[150] Ipsen, I.C.F. (1984)
Singular Value Decomposition with Systolic Arrays.
NASA Technical Report NASA-CR-172396, Langley Research Center, Hampton, Va., July 1984.

[151] Schabowski, R.S., Gai, E., et al. (1984)
Evaluation methodologies for an advanced information processing system.
In *Proceedings of Digital Avionics Systems Conference* (Baltimore, Md., Dec. 3–6, 1984), 217–224.

[152] Dosh, M.K., and Yakos, M.D. (1986)
GPS/INS integration using decoupled residual data.
In *Proceedings of the National Technical Meeting of the Institute of Navigation (ION)* (Long Beach, Calif., Jan. 21–23, 1986), 117–120.

[153] Bierman, G.J., and Belzer, M.R. (1985)
A decentralized squareroot information filter/smoother.
Presented at the 24th IEEE Conference on Decision and Control, Dec. 1985.

[154] Andrisani, D., and Gau, C.F. (1985)
Multistage linear estimation using partitioning.
IEEE Transactions on Automatic Control, AC-30, 2 (Feb. 1985), 183–185.

[155] Kerr, T.H. (1985)
Decentralized filtering and redundancy management failure detection for multisensor integrated navigation systems.
In *Proceedings of the National Technical Meeting of the Institute of Navigation (ION)* (San Diego, Calif., Jan. 15–17, 1985), 191–209.

[156] Hodzic, M., and Siljak, D.D. (1986)
Decentralized estimation and control with overlapping information sets.
IEEE Transactions on Automatic Control, AC-31, 1 (Jan. 1986), 83–86.

[157] Fellerhoff, J.R., Hochrein, G.J., et al. (D.D. Boozer, Ed.) (1985)
SITAN Design for Low-Level Attack Aircraft, vol. I: SITAN Flight Tests, Maps, Algorithm Design, and Performance.
Technical Report AFWAL-TR-84-1148, Sandia National Laboratories, Albuquerque, N. Mex., May 1985.

[158] Jones, H.L., and Pisano, A.L. (1985)
Adaptive Tactical Navigation.
Technical Report AFWAL-TR-85-1015, The Analytic Sciences Corporation, Reading, Mass., Apr. 1985.

[159] Potter, J.E., and Suman, M.C. (1986)
Extension of the midvalue selection technique for redundancy management of inertial sensors.
AIAA Journal on Guidance, Control and Dynamics, 9, 1 (Jan.–Feb. 1986), 37–44.

[160] Hartmann, G.L. (1985)
Fault tolerant hardware/software architectures for flight critical functions.
In AGARD Lecture Series 143: Fault Tolerant Hardware/Software Architecture for Flight Control Function, Oct. 1985.

[161] Papadopoulous, G.M. (1985)
Redundancy management of synchronous and asynchronous systems.
In AGARD Lecture Series 143: Fault Tolerant Hardware/Software Architecture for Flight Critical Function, Oct. 1985.

[162] Anderson, T., and Barrett, P.A. (1985)
Software fault tolerance experiments.
In AGARD Lecture Series 143: Fault Tolerant Hardware/Software Architecture for Flight Critical Function, Oct. 1985.

[163] Heimerdinger, W.L. (1985)
Multi-computer fault tolerant systems using Ada.
In AGARD Lecture Series 143: Fault Tolerant Hardware/Software Architecture for Flight Control Function, Oct. 1985.

[164] Hartman, G.L., Wall, J.E., and Rang, E.R. (1985)
Design validation of fly-by-wire flight control systems.
In AGARD Lecture Series 143: Fault Tolerant Hardwrae/Software Architecture for Flight Critical Function, Oct. 1985.

[165] Belmont, H.H. (1985)
Digital fault-tolerant flight actuation systems.
In AGARD Lecture Series 143: Fault Tolerant Hardware/Software Architecture for Flight Critical Function, Oct. 1985.

[166] McSharry, M.E. (1983)
Fault tolerant flight control avionics integration using MIL-STD-1553B.
Proceedings of the Fifth Digital Aviation Systems Conference (DASC) (Seattle, Wash., Oct. 31–Nov. 3, 1983).

[167] Merril, W.C. (1985)
Sensor failure detection for jet engines using analytical redundancy.
AIAA Journal of Guidance, Control, and Dynamics, 8, 6 (Nov.–Dec. 1985).

[168] Miranker, G., Tang, L., and Wong, C.K. (1983)
A "zero-time" VLSI sorter.
IBM Journal of Research and Development, 27, 2 (Mar. 1983), 140–148.

[169] Leighton, T. (1985)
Tight bounds on the complexity of parallel sorting.
Report MIT/LCS/TM-272 for ONR under DARPA/DoD N00014-80-C-0622, Massachusetts Institute of Technology, Cambridge, Mar. 1985.

[170] Kasif, S. (1984)
Parallel Searching and Merging on ZMOB.
Technical Report CAR-TR-64/CSC-TR-1405, University of Maryland, College Park, Md., June 1984.

[171] Chapiro, D.M. (1984)
Globally-asynchronous locally-synchronous systems.
Ph.D. Dissertation, Department in Computer Science, Stanford University, Stanford, Calif., AD-A154624, Oct. 1984.

[172] Pradhan, D.K. (1985)
Fault Tolerant Computer Research—Interim Scientific Report: Jan. 15, 1984 to Jan. 14, 1985.
Report AFOSR-TR-85-0368, University of Massachusetts, Amherst, Mass., Mar. 4, 1985.

[173] George, A., Heath, M.T., and Lin, J. (1985)
Parallel Cholesky Factorization on a Multiprocessor.
Report ORNL-6124, University of Waterloo and York University, Ont., Canada, under DOE and AFOSR Contract AFOSR-ISSA-84-00056, Mar. 1985.

[174] Basseville, M., Benveniste, A., et al. (1985)
Detection and Diagnosis of Changes in the Eigenstructure of Nonstationary Multivariable Systems.
IRISA Publication 276, Laboratories De Rennes, Dec. 1985.

[175] Rouge, A., Basseville, M., Benveniste, A., and Moustakides, G. (1985)
Optimum Robust Detection of Changes in the AR Part of a Multivariable ARMA Process.
IRISA Publication 277, Laboratories De Rennes, Dec. 1985.

[176] MIL-STD (1983)
Military Standard: Procedures for Performing a Failure Mode Effects and Criticality Analysis.
Report MIL-STD-1629A, Notice 1, June 7, 1983.

[177] Stuzinski, W.A. (1983)
Performance model for square law detectors followed by accumulators and an ORing device.
IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-31, 3 (June 1983), 759–761.

[178] Nuttall, A.H. (1975)
Signal-to-Noise Ratio Requirements for Greatest-of Device followed by Integrator.
Technical Memorandum TC-13-75, Naval Underwater Systems Center, New London, Conn., July 24, 1975.

[179] Hughes Aircraft (1968)
Loss of *S/N* Ratio in Peak Detecting *N* Channels of Data.
Technical Note, Hughes Aircraft Co., Fullerton, Calif., Feb. 12, 1968.

[180] Lull, J.M., Morse, F.L., and Cook, R. (1980)
Coordinate transformation in PLRS.
In *Proceedings of 1980 IEEE Position Location and Navigation Symposium (PLANS)* (Atlantic City, N.J., 8–11, Dec. 8–11, 1980), 335–338.

[181] Fincke, W.H., and Kleinman, A.J. (1980)
Maneuver dependent component error models and synchronized reset filters for inertial navigation systems.
In *Proceedings of 1980 IEEE Position Location and Navigation Symposium (PLANS)* (Atlantic City, N.J., Dec. 8–11, 1980), 107–114.

[182] Smead, F.W. (1983)
Utilizing basic ICNIA capabilities to improve mission availability and reduce pilot workload.
In *Proceedings of the IEEE/AIAA 5th Digital Avionics Systems Conference* (Oct. 31–Nov. 3, 1983), no. 9.5.1.

[183] Mayo, R.A. (1979)
Relative quaternion state transition relation.
AIAA Journal of Guidance and Control, 2, 1 (Jan.–Feb. 1979), 44–48.

[184] Krisnamurti, G., Harshbarger, S.A., and Smith, T.N. (1985)
The design and performance of GPS Phase III user equipment navigation software.
In *Proceedings of the National Technical Meeting of the Institute of Naviation (ION)* (San Diego, Calif., Jan. 15–17, 1985), 101–109.

[185] Hudak, G. (1986)
GPS user equipment, an evolutionary assessment.
In *Proceedings of the National Technical Meeting of the Institute of Navigation (ION)* (Long Beach, Calif., Jan. 21–23, 1986), 5–14.

[186] Cumming, R.C., Heinze, R.L., and Moser, R.R. (1986)
GPS/JTIDS compatibility.
In *Proceedings of the National Technical Meeting of the Institute of Navigation (ION)* (Long Beach, Calif., Jan. 21–23, 1986), 35–40.

[187] Kerr, T.H. (1985)
The proper computation of the matrix pseudoinverse and its impact in MVRO filtering.
IEEE Transactions on Aerospace and Electronic Systems, AES-21, 5 (Sept. 1985), 711–724.

[188] Lupash, L.O. (1985)
A new algorithm for the computation of the geodetic coordinates as a function of Earth-centered Earth-fixed coordinates.
Journal of Guidance, Control, and Dynamics, 8, 6 (Nov.–Dec. 1985), 787–789.

[189] Sullivan, N., and Munson, S. (1986)
Final Report: Preliminary Definition Study for a Fault-Tolerant Inertial Reference System, vol. I.
ARINC Research Publication 2922-02-1-3917, Jan. 1986.

[190] Vaman, H.J. (1985)
Optimal online detection of parameter changes in two linear models.
Stochastic Processes and Their Applications, 20 (1985), 343–351.

[191] Gobzemis, A.Y., and Kizub, V.A. (1985)
Use of the apparatus of petri nets in the solution of technical diagnostic problems: A survey.
Automatic Control and Computer Science (Russian Journal in English), *19*, 1 (1985), 19–25.

[192] Pradhan, D.K. (Ed.) (1985)
Fault Tolerant Computing: Theory and Techniques.
Englewood Cliffs, N.J.: Prentice-Hall, 1985.

[193] Isermann, R. (1984)
Process fault detection based on modeling and estimation methods—A survey.
Automatica, 20, 4 (1984), 387–404.

[194] Yount, L.J. (1985)
Generic fault-tolerance techniques for critical avionics systems.
In *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Snowmass, Colo., Aug. 19–21, 1985), loose handout.

[195] Driscoll, K.R. (1985)
Reversion, A response to generic faults.
In *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Snowmass, Colo., Aug. 19–21, 1985), loose handout.

[196] Handelman, D.A., and Stengel, R.F. (1985)
Combining quantitative and qualitative reasoning in aircraft failure diagnosis.
In *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Snowmass, Colo., Aug. 19–21, 1985), 366–375.

[197] Chin, H.H. (1985)
Distributed Kalman filter in an integrated SAHRS/GPS navigation system.
In *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Snowmass, Colo., Aug. 19–21, 1985), 209–222.

[198] Morse, A.S. (1986)
State estimation using distributed processing.
In *Proceedings of the 25th Conference on Decision and Control* (Patras, Greece, Dec. 1986).

[199] Hepburn, J.S.A., Reid, D.B., Widnall, W.S., et al. (1984)
Motion compensation for high resolution spotlight SAR.
In *Proceedings of Position Location and Navigation Symposium (PLANS)* (San Diego, Calif., Nov. 26–29, 1984).

[200] Farrell, J.L. (1985)
Strapdown inertial navigation system requirements imposed by synthetic aperture radar.
AIAA Journal of Guidance, Control, and Dynamics, 8, 4 (July–Aug. 1985), 433–439.

[201] Smith, T.B. (1981)
Synchronous fault-tolerant flight control systems.
Proceedings of AIAA Computers in Aerospace III Conference (San Diego, Calif., Oct. 26–28, 1981), 55–60.

[202] Gaston, G.G., Costlow, L.E., and Perry, J.R. (1984)
A highly monitored digital flight control system for the AV-88 Harrier II.
In *Proceedings of AIAA Digital Avionics Systems Conference* (Baltimore, Md., Dec. 3–6, 1984).

[203] Ludvigson, M.T., and Milton, K.L. (1985)
High-speed bus for Pave Pillar applications.
In *Proceedings of AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference* (Long Beach, Calif., Oct. 21–23, 1985), 75–83.

[204] Gupta, N.K., and Walker, R.A. (1984)
Robust fault detection techniques.
In *Proceedings of the AIAA Guidance and Control Conference* (Seattle, Wash., Aug. 20–22, 1984), 758–760.

[205] Alexandro, F.J., and Tsou, J. (1984)
Instrument failure detection and isolation in a system with variable plant parameters.
In *Proceedings of the AIAA Guidance and Control Conference* (Seattle, Wash., Aug. 20–22, 1984), 176–181.

[206] Osder, S.S. (1982)
Generic faults and design solutions for flight-critical systems.
In *Proceedings of the AIAA Guidance and Control Conference* (Seattle, Wash., Aug. 9–11, 1982), 509–518.

[207] Moldovan, D.I. (1983)
On the design of algorithms for VLSI systolic arrays.
Proceedings of the IEEE, 71, 1 (Jan. 1983), 113–120.

[208] Rao, S.K., Jagadish, H.V., and Kailath, T.
Analysis of interative algorithms for multi-processor implementations.
To appear.

[209] Jagadish, H.V., Rao, S.K., and Kailath, T.
Multi-processor architectures for interative algorithms.
To appear.

[210] Kolata, G. (1986)
Faster division with computers.
Science, 233, 4761 (July 18, 1986), 281.

[211] Kerr, T.H. (1982)
A Conservative View on the GLR Failure and Event Detection Approach.
Intermetrics Internal Memo, Mar. 1982; available from author upon request.

[212] Kerr, T.H., and Chin, L. (1980)
A stable decentralized filtering implementation for JTIDS RelNav.
In *Proceedings of Position Location and Navigation Symposium (PLANS)* (Atlantic City, N.J., Dec. 8–11, 1980), 318–328.

[213] Kerr, T.H.
Testing matrices for positive definiteness and semidefiniteness and application examples that spawn the need.
AIAA Guidance, Control, and Dynamics, 10, 2, (Mar.–Apr. 1987).

[214] Massoumnia, M-A. (1986)
A geometric approach to failure detection and identification in linear systems.
Doctor of Science Thesis, Space Systems Laboratory of the Department of Aeronautics and Astronautics, Report SSL 5-86, Massachusetts Institute of Technology, Cambridge, Feb. 1986.

[215] Brown, R.G., and Hwang, Y.C. (1986)
GPS failure detection by autonomous means within the cockpit.
In *Proceedings of the 42nd Annual Meeting of the Institute of Navigation (ION)* (Seattle, Wash., June 24–26, 1986), paper 2.

[216] Ben-Haim, Y.
Convexity analysis and optimum malfunction isolation.
Automatica. To appear.

[217] Merrifield, J.T. (1986)
Ford developing expert system to aid satellite fault repair.
Aviation Week and Space Technology (May 12, 1986), 89–95.

[218] Kleinman, K.L., and Kleinman, A.K. (1974)
Computational methods for the steady-state Kalman filter in high order navigation models.
In *Proceedings of the 12th Annual Allerton Conference on Circuit and System Theory* (P.V. Kokotovic and E.S. Davidson, Eds.) (Monticello, Ill., Oct. 2–4, 1974).

[219] Sandberg, H.J., and Dushman, A. (1973)
System design and analysis of ESGM/SINS navigation systems.
In *Proceedings of the Conference of Institute of Navigation* (Marine meeting) (Kings Point, N.Y., 1973).

[220] Ludvigson, M.T., and Milton, K.L. (1985)
High-speed bus for Pave Pillar applications.
In *Proceedings of the AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference* (Long Beach, Calif., Oct. 21–23, 1985), 75–83.

[221] Klos, L.C., and Hewell, J.S. (1985)
Transparent contention buses can improve architectures.
In *Proceedings of AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference* (Long Beach, Calif., Oct. 21–23, 1985), 64–74.

[222] Wax, M., and Kailath, "Decentralized Processing in Sensor Arrays," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 4, pp. 1123–1129, October 1985.

[223] Tenney, R.R., and DeLaney, J.R., "A Distributed Aeroacoustic Tracking Algorithm," *Proceedings of American Control Conference*, San Diego, CA, pp. 1440–1450, 6–8 June 1984.

[224] Maeda, H., Ohta, Y., et al., "Fault Diagnosis of Non-Linear Systems: Graphical Approach to Detectability, Distinguishability, and Diagnosis Algorithm," *Circuit Theory and Applications*, Vol. 14, pp. 195–209, 1986.

[225] "Panel Sees U.S. Falling Behind in Supercomputer Technology," *Aviation Week & Space Technology*, pp. 71–72, 29 Aug. 1983.

[226] Special Issue: Supercomputers—Their Impact on Science and Technology, *Proceedings of IEEE*, Vol. 72, No. 1, Jan., 1984.

[227] "The Cray X-MP Series of Computer Systems," Cray Research, Inc., Brochure No. MP-0102A, 1986.

[228] Morf, M., Levy, B., and Kailath, T.
"Square-Root Algorithms for the Continuous-Time Linear Least Squares Estimation Problem."
Proceedings of the 1977 IEEE Conference on Decision and Control, New Orleans, LA, pp. 944–947, 7–9 Dec. 1977.

[229] Schmidt, D.J.
"Discussion of 'Square Root Algorithms for Parallel Processing in Optimal Estimation'."
Automatica, Vol. 15, pp. 497–499, 1979.

[230] Widnall, W.S., and Bryson, A.E.
"Comment on 'Optimizing the Gains of the Baro-Inertial Vertical Channel."
AIAA Journal of Guidance and Control, Vol. 3, No. 3, pp. 286–288, May–June 1980.

[231] Cannon, M.J., and Hollaar, L.A.
"Alternative Approach to Multisensor Navigation."
AIAA Journal on Guidance and Control, Vol. 3, No. 5, pp. 398–404, Sept.–Oct. 1980.

[232] Rader, C.M., and Steinhardt, A.O.
"Hyperbolic Householder Transformations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 6, pp. 1589–1602, Dec. 1986.

[233] Schweppe, F.C.
Uncertain Dynamic Systems, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[234] Reif, J.H., and Tyqar, J.D.
"Efficient Parallel Pseudo-Random Number Generation,"
Advances in Cryptology—Crypto'85.
Lecture Notes in Computer Science, Springer-Verlag, N.Y., 1986.

[235] Churchyard, J.N.
"Comments on 'A New Algorithm for the Computation of the Geodetic Coordinates as a function of Earth-Centered Earth-Fixed Coordinates'."
AIAA Journal of Guidance, Control, and Dynamics, Vol. 9, No. 4, July–Aug. 1986.

[236] Bertsekas, D.P., Tsitsiklis, J.N., and Athans, M.
"Convergence Theories of Distributed Iterative Processes: A Survey."
Stochastic Programming, Eds. M. Thoma and A. Wyner, Lecture Notes in Control and Information Sciences Series No. 76, Springer-Verlag, N.Y., 1986.

[237] Telksyns, L. (Ed.)
"Detection of Changes in Random Processes."
Optimization Software, Inc., Publications Division, N.Y., 1986.

[238] Lala, J.H.
"Fault Detection, Isolation, and Reconfiguration in the Fault Tolerant Multiprocessor."
AIAA Journal of Guidance, Control, and Dynamics, Vol. 9, No. 5, pp. 583–592, Sept.–Oct. 1986.

[239] Sporn, S.R.
"Comment on 'Automated VOR Ground Check Techniques."
IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-22, No. 2, pp. 214–215, Mar. 1986.

[240] Robinson, P.N.
"Further Corrections to 'Depth of Field for SAR with Aircraft Acceleration'."
IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-22, No. 4, pp. 468–469, July 1986.

[241] Heller, D.
"A Survey of Parallel Algorithms in Numerical Linear Algebra."
SIAM Review, Vol. 20, No. 4, pp. 740, Oct. 1978.

[242] Kerr, T.H.
"Computational Techniques for the Matrix Pseudoinverse in Minimum Variance Reduced-Order (MVRO) Filtering."
Advances in Control and Dynamic Systems, Vol. XXVIII: Advances in Algorithms and Computational Techniques for Dynamic Control Systems, C.T. Leondes (Ed.), Academic Press, Orlando, FL, 1987.

Thomas Kerr (S'67—M'74—SM'85) was born in Washington, D.C. on November 9, 1945. He received the B.S.E.E. (magna cum laude) from Howard University, Washington, D.C. in 1967 and the M.S. and Ph.D. degrees (via NSF) in the electrical engineering specialty of control/estimation from the University of Iowa, Iowa City, IA in 1969 and 1971, respectively.

A staff member at MIT Lincoln Laboratory since October 1986 in the area of spectral estimation (which uses the same estimation/detection/identification/pattern recognition tools) for radar tracking applications, he had been a senior analyst/systems engineer at Intermetrics from 1979 with interests in navigation system and sonobuoy system tracking/implementation/Kalman filtering/failure detection. His previous experience includes university teaching/research (1967–1971), analysis, computer simulation, and real-time implementation at General Electric's Corporate R&D Center in Schenectady, N.Y. (1971–1973), and navigation/filtering/failure detection applications at TASC (1973–1979).

His DOD application involvements include: investigations and computer implementations of parameter identification, mathematical and algorithmic aspects of estimation and control, decentralized and nonlinear filtering, and quantifying associated computer burdens, hypothesis testing/failure detection, effects of failure detection on the reliability/availability of reconfigurable modular composite systems, bicriteria (Pareto-) optimization algorithms, and procedures for Kalman filter sensor selection and scheduling, tractable applications of level-crossing theory, point-process detection in antisubmarine warfare, optimal search and screening techniques, multi-target tracking, and pattern recognition. These above investigations were performed for various applications including Poseidon and Trident SINS/ESGN navigation systems, JTIDS RelNav, ICNIA and other GPS-aided multi-sensor avionics navigation applications, minesweeper PINS, PTA sonobuoy tracking, Post-Coherence Function sonobuoy target tracking, ASW search, submarine antenna detectability to radar surveillance, and helicopter-based Missile Warning System development and refinement.

Dr. Kerr is also a member of Tau Beta Pi, Eta Kappa Nu, Sigma Pi Sigma, Pi Mu Epsilon, AIAA, ION, AAAS, the Naval Institute, ADPA, NSIA, and IEEE AC, IT, AES, and ASSP groups.