THE JOURNAL OF NAVIGATION (2005), **58**, 405–417. © The Royal Institute of Navigation
doi:10.1017/S0373463305003383 Printed in the United Kingdom

A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems

David J. Allerton

(*The University of Sheffield*)
(Email: d.j.allerton@sheffield.ac.uk)

Huamin Jia

(*Cranfield University*)

This paper reviews currently existing fault-tolerant navigation system architectures and data fusion methods used in the design and development of integrated aircraft navigation systems and also compares their advantages and disadvantages. Four fault-tolerant navigation system architectures are reviewed and the associated Kalman filter architectures and algorithms are discussed. These techniques have been used in most integrated aircraft navigation systems. The aim of this review paper is to provide a guide for navigation system designers to develop future aircraft multisensor navigation systems.

KEY WORDS

1. Data Fusion. 2. Fault Tolerance. 3. Kalman Filter Architectures.

1. INTRODUCTION.

Aircraft navigation systems are a flight-critical system and must be designed to meet the required navigation performance (RNP) requirements for safety, in terms of accuracy, integrity, continuity and availability. In the past decade, various forms of fault-tolerant aircraft navigation systems have been developed to achieve these requirements. This paper reviews the methodologies used in the design and development of fault-tolerant aircraft navigation systems from the perspective of system design. In Section 2, conventional fault-tolerant navigation system architectures are outlined and briefly compared. Section 3 reviews the development of several Kalman filter architectures and filtering algorithms. The methods summarised in this paper have been widely applied in many integrated aircraft navigation systems and enable navigation system developers to design and develop future aircraft multisensor navigation systems.

2. FAULT-TOLERANT NAVIGATION SYSTEMS.

Fault-tolerant navigation systems have been in use for over 30 years. The design methods incorporate both fault-tolerant strategies and data fusion techniques to enhance

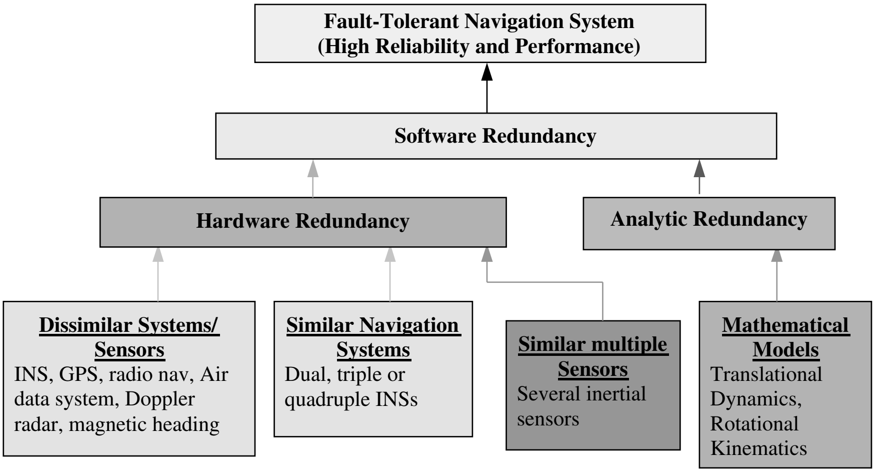

Figure 1. Heirarchical structure of fault-tolerant design methods.

reliability and safety and also to improve the performance of aircraft navigation system in terms of the RNP parameters. During this development, three redundancy strategies have been proposed: hardware redundancy, software redundancy and analytical redundancy. Hardware redundancy uses multiple navigation sensors/systems to achieve fault tolerance and improve the performance of an aircraft navigation system. This approach is based on the principle that measurements from various sensor systems are independent, redundant, complementary or cooperative. These different types of measurements can be combined by means of data fusion algorithms, so that the overall system performance is better than that of each individual system. Hardware redundancy techniques have been widely applied to many avionics systems[1,2,3]. Software redundancy uses different software versions to increase the safety and integrity of navigation solutions by avoiding possible errors caused by software and computing failures. However, software redundancy cannot increase the accuracy of navigation solutions. Analytical redundancy is based on the knowledge of rotational kinematics and translational dynamics of an aircraft to enhance hardware redundancy[4]. Analytical redundancy is generally used to generate additional redundant information for the diagnosis of system failures rather than the improvement of accuracy of navigation systems[5]. For that reason, analytical redundancy is considered as a failure detection method in most practical systems. Figure 1 outlines the fault-tolerant design methods used for aircraft navigation systems. Hardware redundancy plays an essential role in the design of fault-tolerant navigation systems and the level of fault tolerance depends on both the architecture of hardware redundant systems and the data fusion methods implemented. Three types of hardware redundancy have been developed for the design of fault-tolerant aircraft navigation systems: system-level redundancy, sensor-level redundancy and distributed redundancy.

2.1. *System-Level Redundancy Architecture*. A typical system-level redundancy architecture is shown in Figure 2, where each Inertial Navigation System (INS) operates independently and there is no data communication between these systems. This is generally known as an independent system architecture. Each INS can also be

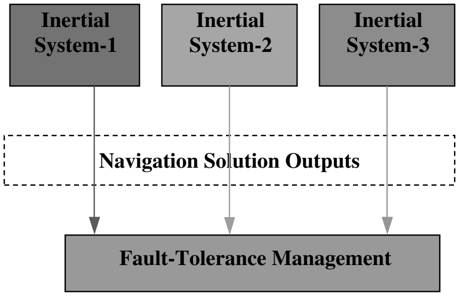

Figure 2. System-level redundancy.

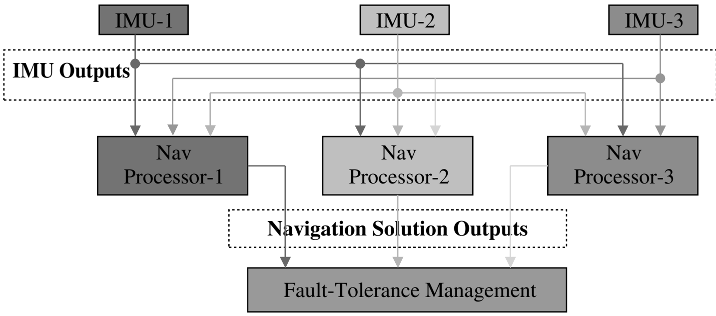

Figure 3. IMU-level redundancy.

integrated with other navaid systems to improve the navigation accuracy and to control the growth of inertial sensor errors with time. Fault-tolerant methods used to check the consistency and failures of all the INSs are typically majority-voting methods or weighted-mean methods. In order to achieve fail-operational/fail-safe operation, at least three INSs are needed in this configuration. In other words, at least nine pairs of inertial sensors (accelerometers and gyros) are needed, where each INS is a conventional orthogonal configuration. The main advantage of this architecture is that the design and integration is simple and does not require complex fault-tolerant methods for the diagnosis of system failures. However, if any sensor in one INS fails, then this INS has to be removed from the fault-tolerant architecture. Consequently, this architecture cannot exploit the benefits of redundant inertial sensors to dynamically reconfigure an aircraft navigation system when one INS fails. This traditional redundant architecture is still used in many existing avionic systems[6], although these systems are expensive and the duplication of INS modules can result in a significant increase in mass.

2.2. *Sensor-Level Redundancy Architectures*. Sensor-level redundant architectures were developed with the advent of high-speed, large memory embedded microprocessors and low-cost, small-size and low-mass inertial measurement units (IMU). Several redundant schemes have been proposed, including IMU-level and multisensor redundancies.

2.2.1. *IMU-Level Redundancy*. An IMU-level redundant architecture used in many aircraft navigation systems is shown in Figure 3, where duplex or triplex

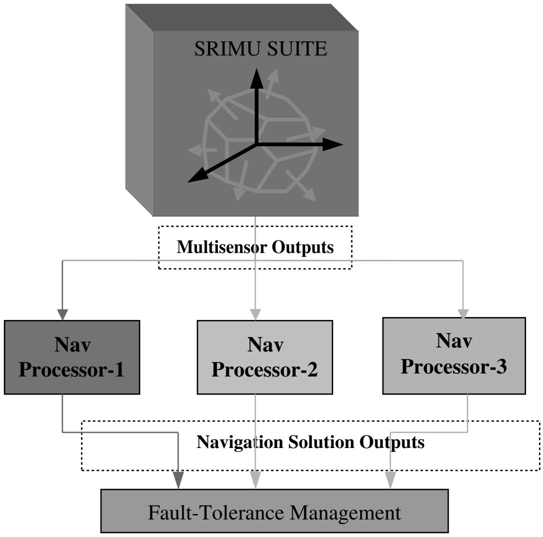

Figure 4. Multisensor redundancy.

conventional IMUs are configured in a federated architecture to obtain fault tolerance. Each IMU can be skewed with respect to the aircraft body axes when it is mounted in the aircraft in order to reduce the number of IMUs[7]. Theoretically, a fault-tolerant navigation system consisting of two IMUs affords fail-operational/fail-operational/fail-safe operation if one of these IMUs is skewed relative to the aircraft body axes, or is a non-orthogonal IMU. In this configuration, six pairs of inertial sensors can achieve a higher level of fault tolerance in comparison with three independent INSs. Each navigation processor can combine the outputs of all IMUs with data from aiding systems to estimate the aircraft motion states and to perform sensor failure detection and isolation and system reconfiguration. Compared with the INS-level redundancy, this architecture significantly increases the level of fault tolerance and makes effective use of existing IMU equipment. However, resultant fault-tolerant systems still share some of the disadvantages of system-level redundant architectures and considerable efforts are being made to reduce volume, weight and cost.

2.2.2. *Multisensor Redundancy* An alternative development is to integrate multiple inertial sensors in a single suite in the form of non-orthogonal configurations[3], known as skewed redundant IMU (SRIMU) configurations. One multisensor suite can thus replace multiple IMUs to reduce the volume, weight and power required for an aircraft navigation system. A representative architecture of multisensor fault-tolerant systems is shown in Figure 4, where the multisensor suite is a dodecahedron configuration. Six pairs of inertial sensors are installed perpendicular to the parallel faces of a regular dodecahedron. The SRIMU outputs are sent to redundant navigation processors, each individually performing the navigation and attitude computations, sensor FDI functions and navigation system reconfiguration. The multisensor redundancy is a cost-effective approach that exploits the benefits of emerging inertial sensor technologies and high-speed embedded microprocessor systems. Multisensor technology will provide the basis for the future generations of fault- tolerant navigation systems.

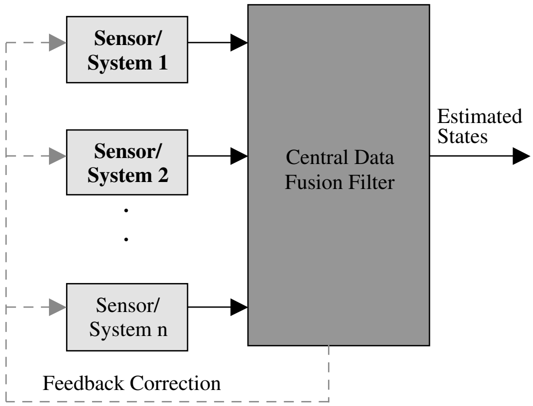

Figure 5. Centralised data fusion architecture.

2.3. *Distributed Redundant Architectures.* Distributed redundant architectures is a new fault-tolerant concept which has been developed with the introduction of distributed and integrated modular avionics architectures. A current combat platform may have a total of twelve traditional IMUs of various quality, providing the state vector information required by avionic systems and weapon systems[8]. In this architecture, inertial sensor systems are mounted at several locations in an aircraft, not only to meet the fault tolerance requirements of navigation systems, but also to provide accurate local inertial vector states for other systems, for example, weapon control systems and imaging sensors and to provide radar stabilization and motion compensation. The concept of using an inertial network for aircraft avionics was initially proposed by Kelley, Carlson and Berning[8] in 1994. However, no research has been published describing a systematic study of inertial network architectures for fault tolerant aircraft navigation systems, in terms of combining data fusion methods, dynamic alignment and correction of distributed inertial sensor systems and distributed sensor failure detection and isolation techniques.

3. DATA FUSION FILTER ARCHITECTURES. Kalman filtering techniques have been developed for applications in aircraft navigation, control and guidance since the 1970s. During this period, many Kalman filter architectures and filtering algorithms have been proposed as prime data fusion methods, to combine multiple navigation sensors/systems to achieve the required navigation performance. Data fusion filter architectures currently used in aircraft integrated navigation systems can be categorised as four types: centralised, cascaded, federated and distributed data fusion architectures.

3.1. *Centralised Filter Architecture.* The centralised filter architecture is illustrated in Figure 5, where measurements or data from all navigation sensors/systems are processed in a central data fusion filter to obtain the accurate estimates of aircraft motion states. This architecture is the most common filter design implemented in current integrated navigation systems, including INS/GPS/Doppler integrated systems[9], GPS/Doppler integrated systems[10] and tightly-coupled GPS/inertial systems[11,12]. In these systems, INS outputs and raw GPS measurements are combined in a centralised filter to estimate the navigation state errors and sensor errors, including

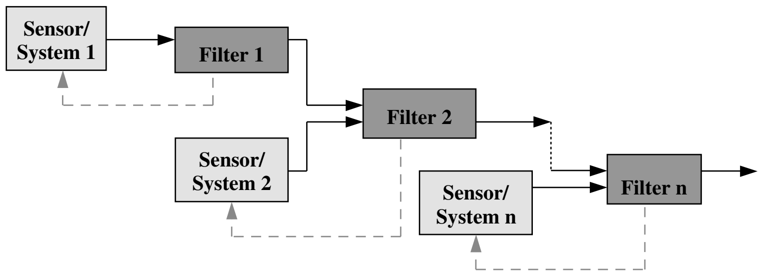

Figure 6. Cascaded data fusion architecture.

GPS receiver clock errors, inertial sensor errors and baro-altimeter errors. Numerous covariance analysis methods and numerical computations of the standard and extended Kalman filters have been reported in the literature. Theoretically, the centralised filter can obtain optimal estimates of the aircraft motion states. However, with the increasing number of sensor systems in aircraft, the filtering algorithms can be quite complex and the centralised filter computation can be time-consuming as a result of the large number of states in the dynamic models of the filter. Accordingly, the centralised filter is not necessarily an appropriate methodology in the development of fault tolerant multisensor navigation systems[20,29,31]. To overcome the limitations of the centralised filter, other filter architectures have been developed.

3.2. *Cascaded Filter Architecture*. The cascaded filter architecture is shown in Figure 6, where the outputs of one filter are used as inputs to a subsequent filter stage. The filter outputs include the estimates of the system states and their error covariances. This filter architecture has been proposed for the integration of existing navigation systems which contain their own Kalman filters. The cascaded filter can improve the accuracy of integrated navigation systems and also perform in-flight calibration or transfer alignment between an INS/GNSS integrated system and a slave INS or attitude heading reference system (AHRS). This architecture has been used in GPS/INS/terrain-aided navigation systems[13] and loosely-coupled GPS/INS integrated navigation systems, where the GPS navigation solutions, derived by an GPS internal filter and INS data, are combined in a separate cascaded filter external to the GPS receiver to estimate the navigation state errors and the inertial sensor errors. The GPS filter estimates the GPS receiver clock errors. However, the GPS filter is generally based on a simplified model and may not output the computed error covariances. Consequently, the cascaded filter may not have access to covariance information. Schlee et al[14] developed a cascaded filtering algorithm to improve the accuracy of an existing GPS/inertial system, known as a master INS, which utilises an internal filter to estimate the master INS navigation solutions and the GPS clock errors. This cascaded algorithm also provides transfer alignment between the master INS and a second inertial system. Their study showed that improvement in the accuracy of the master INS and the accuracy of the transfer alignment depend on the update rate of the cascaded filter. However, correlations of the state errors caused by the internal filter are ignored in the measurement noise matrix of the cascaded filter. From Kalman filter theory, the non-diagonal elements of the state error covariance matrix of the filter (which represent the correlations) can only be ignored if the filter offers highly accurate estimates of the navigation states and the magnitudes of the off-diagonal elements are far less than the diagonal elements. Otherwise, the performance of the cascaded filter may be degraded as a result of the correlation.

Wade and Grewal[15] analysed the effect of this correlation on the accuracy of cascaded GPS/INS systems; their results show that the accuracy of cascaded systems depends on the correlation matrix. When the state errors estimated by the internal filter are closely correlated, the cascaded filter may incorrectly estimate the navigation state errors and the inertial sensor errors. Wade and Grewal suggest adjusting the measurement noise matrix by using adaptive process noise in the cascaded filter. However, development of this adaptive process and identification of the measurement noise matrix are not reported in detail.

In order to improve the robustness of the cascaded filter to input conditions and adverse environments, Karatsinides[16] proposes two methods for dealing with the GPS position bias and identifying the statistical values of measurement noise for the cascaded filter. The GPS positioning solution contains biases resulting from satellite clock errors, ephemeris errors, ranging signal propagation delay and geometries of visible satellites. Although GPS position bias is unobservable and cannot be estimated in the GPS filter, it can influence the accuracy of cascaded GPS/INS systems through the error covariance matrix. The first method models GPS position bias as a first-order Gauss-Markov process and then uses these biases as the *consider-states* of a Schmidt-Kalman filter. The part of the Schmidt-Kalman gain matrix related to the *consider-states* is set to zero in order to ignore the estimated *consider-states*. The second method computes the variances and covariances of the errors of the navigation states derived by the GPS filter, using conventional computation equations of variance and covariance, provided that the update rate of the cascaded filter is less than the GPS filter.

The cascaded filter architecture can be used to integrate existing navigation systems into a fully integrated system and may only require minimal modifications to existing navigation systems. In practice, most existing navigation systems do not output covariance data of the navigation state errors. Consequently, the cascaded filter is extremely dependent upon the methods that are used to estimate the covariances of the primary filter and the performance of the primary filter. Moreover, tuning of the primary filter is of critical importance to the performance of the cascaded filter[15].

3.3. *Federated Filter Architecture*. The federated filter architecture was initially recommended by Carlson[17] for integrating multiple navigation sensor systems in order to provide a high level of fault tolerance and accuracy. This is a two-stage filtering architecture, as shown in Figure 7, where all the parallel local filters combine their own sensor systems with a common reference system, usually an inertial system, to obtain the local estimates of the system states. These local estimates are subsequently fused in a master filter to achieve the global estimations. By using a common reference system, all parallel filters have a common state vector. The federated filter is generally designed on the basis of two different strategies[17,18]. In the first method, the local filters are designed independent of the global performance of the federated filter and estimate n sets of local state vectors and their associated covariances by using their own local measurements. These n sets of the local state estimates are then weighted by their error covariances to obtain the global state estimates. The second method is based on the global optimality of the federated filter; the local filters are derived from the global model of the federated filter and estimate n versions of the

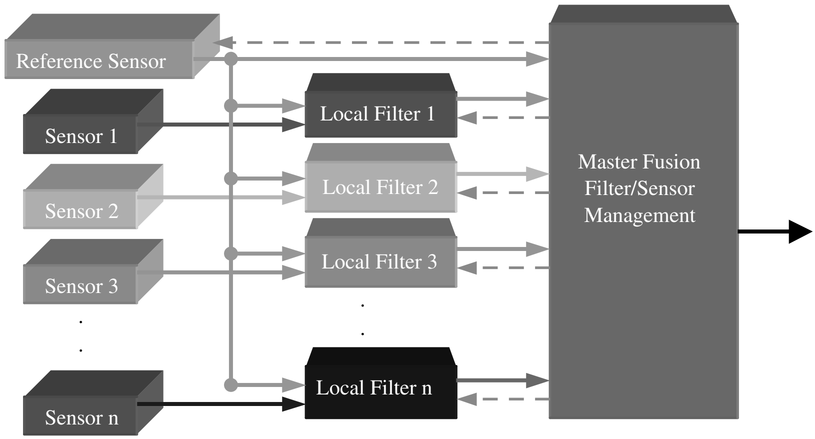

Figure 7. Federated data fusion architecture.

global states from local sensor measurements. These n versions of estimates are weighted by their error covariances to obtain the global optimality. The master filter is a weighted least-squares estimator. Carlson[19] developed a square-root form of the federated filtering algorithm to increase the computational precision and the numerical stability of the federated filter.

A significant feature of the federated filtering process is that a reference INS must be used to create the common system states in the local and master filters, which are the navigation states. Therefore, each local filter can obtain the suboptimal navigation states. A comparison of the federated and centralised filters has shown that the federated architecture offers improvements in failure detection, isolation and recovery (FDIR) and fault tolerance over the centralised filter[20].

Levy[21] uses dual state suboptimal analysis to model the true world state vector and develops covariance analysis algorithms for assessing the sub optimality of both the cascaded and the federated filters. The dual state contains the states of the first and second filters in the case of the cascaded filter (or the states of all parallel filters and the master filter in the case of the federated filter). Levy's results have shown that the cascaded and federated filters are seldom optimal in comparison with the centralised Kalman filter. As the master filter updates become sparser, the actual performance of the federated filter degrades in comparison with the centralised filter. The federated filter is only optimal (or equivalent to the centralised filter) when the full global state is modelled in each local filter and the master filter is run at the update rate of the local filters.

Tupysev[22] develops a federated filtering algorithm based on the principles of state vector augmentation and the rejection of partial information. Unlike Carlson's filter, the global state model that is used to derive the parallel local filters contains a common state vector plus individual local bias state vectors instead of all the states of the local filters.

However, the use of a reference navigation system as a common information source of all local filters in the federated filter architecture means that common mode failures in the reference system can corrupt the performance of these filters. This influence can further degrade the level of fault tolerance and FDIR functions.

This problem seems to have been ignored in current designs of federated integrated navigation systems. The federated filter has been applied to several multisensor navigation systems, for example, GPS/INS/SAR/terrain aided navigation and tracking systems[23]. It should be noted that the federated filter is sometimes referred to as the decentralised filter[27].

3.4. *Distributed Filter Architectures.* Distributed filter architectures were originally developed for target tracking and identification where distributed sensor systems (possibly in different platforms) are combined in order to estimate and identify various moving targets in military applications. Liggins et al[24] gives a comprehensive survey of distributed fusion architectures for target tracking. Distributed filtering techniques used for the design and development of fault-tolerant navigation systems have appeared since 1990[27]. The cascaded and federated filter algorithms are special cases of the distributed filter architectures. Unlike the filter architectures described above, distributed filter architectures have no standard models. In general, there are two main data fusion approaches to the design of distributed filters, known as measurement fusion and state fusion. In state fusion, the local states estimated by the local filters are fused in a central filter to obtain global estimations. By contrast, in measurement fusion, various subsets of all the sensor measurements are fused by means of a bank of Kalman filters to obtain multiple state estimation versions of the global system states, which are combined to obtain the more accurate global state estimation and to detect system failures. However, there may be no central data fusion in a fully distributed multisensor data fusion system. In fact, the distributed filter architecture offers the most flexible scheme in the design of multisensor navigation systems.

Several distributed filtering algorithms have been developed since 1980 for the design of various distributed control systems, target tracking systems and integrated navigation systems. Speyer[25] describes a distributed filtering algorithm in which each of K local filters has its own local sensor measurements and the same state model. Each local filter computes the global estimate of the system state vector. The information shared between these local filters consists of the local estimates and error covariances and an additional (locally computed) data-dependent term, which is a dynamic compensation to account for the correlation between the local estimates. Speyer's filter is a fully distributed filtering architecture and has a high level of fault tolerance. However, by using the same state model, this filtering algorithm cannot be used in a distributed inertial sensor system where the local state vector is needed for a specific application, for example, local motion compensation.

Willsky et al[26] consider a problem where two local filters have models which differ from the global model. Each local filter processes its local measurements and a fusion algorithm (based on the global model) computes a dynamic correlation correction term. The local estimates are then combined to obtain the global estimate. A necessary and sufficient condition for recovering the global state from the local states is that a relationship must exist between the observation matrix of the global state model and that of each local state model. This relationship is formulated as a static matrix transformation. In other words, the local state vector is a subset of the components of the global state vector. This algorithm has been extended to the design of a multisensor navigation system[33]. However, these algorithms imply that both the local and the global states are represented in the same coordinate system and this is not necessarily true for distributed inertial sensor systems.

Kerr[27] proposes a decentralised filtering structure but the decentralised filter algorithms applicable for this structure are not detailed. However, some filtering algorithms, for example, Speyer's parallel filtering algorithms[26], may be used for this decentralised structure. In terms of the filter architecture, Kerr's version is similar to the federated filter architecture given by Carlson[17]. The differences between these architectures are the individual methods used for detection and isolation of subsystem failures. For example, Kerr's filter uses voter/monitoring methods based on Gaussian confidence regions of the estimated states whereas Carlson's filter uses filter residuals to detect sensor and subsystem failures.

Brumback and Srinath[28] describe a distributed filtering mechanism that is a hierarchical filtering architecture, where the local filters fuse different subsets of all measurements for local state estimates and failure detection and isolation. A master filter combines the outputs of failure-free local filters to yield the global estimation. The local filters in the distributed filter architecture can have system models, which are different from the global model.

Hashemipour et al[29] introduce decentralised Kalman filtering algorithms for three types of sensor system networks: sensor collected, time sequential measurements and a hybridisation of these two types. In Hashemipour's filter, each local filter has the same state model as the central filter and the observation matrix of each local model corresponds to one sub-matrix of the observation matrix of the global model. Each local filter computes the global estimation and its local error covariance; these are subsequently fused in a central filter to obtain the global optimal estimation. Accordingly, this filtering algorithm is similar to Speyer's filter, but uses the information form of the Kalman filter and does not need feedback from the central filter to the local filters. Although this algorithm has been applied to target-tracking problems, it is not suitable for distributed multisensor navigation systems because feedback control is an important means to correct sensor errors in a distributed inertial sensor system.

Hong[30] presents a distributed multisensor integration algorithm in which the local measurements, together with previous global estimates obtained via the communication network, are locally processed to obtain the local state estimate and the local error covariance. These local estimates (state and covariance) are fused in a central filter to obtain the global estimate. Because the local state and covariance predictions are derived from the previous global estimates, the local filters have no state models. However, rotation matrices and translation transformations are introduced to define the relationships between the local states and the global (central) state. Moreover, this algorithm was designed to minimise the uncertainties of these transformations. It should be noted that the same relationships are also used for the measurement transformations from the local nodes to the central node. This is not necessarily true in distributed inertial sensor systems, especially when a nonlinear relationship exists between the measurements and the states. Compared with Speyer's filtering algorithm, this method simplifies the complexity of the distributed filtering algorithms. However, the local states greatly depend on the global states because this method lacks local dynamic models.

Roy et al[31] propose a square root filtering structure where parallel local filters have a smaller dimension than the global filter. Paik et al[32] develops a gain fusion algorithm for decentralised parallel Kalman filters to obtain computation-efficient suboptimal estimation results. Raol et al[33] describe a decentralised square-root information filtering scheme where all information fusion is processed locally at each node and there is no central fusion. These algorithms can improve the computational precision and numerical stability of existing distributed filtering algorithms.

Fully distributed filtering architecture and information fusion algorithm are developed, where no central data fusion centre is needed[34]. Each local filter has its own local system model and processes the local measurements and information assimilated from other filters to obtain a global estimate of the system state. However, there is still a key problem to be considered; the dynamic relationship between the local states must be determined, especially if the local state models are different. Berg et al[35] describe the static relation between the local states and the global state using an approach similar to Speyer's method[25].

For aircraft systems, multisensor data fusion offers potential improvements in performance in the following areas of navigation:

- Aircraft navigation system RNP parameters;
- Fault tolerance of navigation systems;
- Estimation of local motion states.

The majority of previous applications have focused on meeting RNP and reliability requirements. In other words, existing distributed filtering algorithms have preserved the global optimality of the navigation states, which is a desirable feature and serves as a benchmark for other avionic systems. However, these methods rarely consider the dynamics of the local subsystems or the dynamic relationships between the local subsystems. Some algorithms still require extensive computations of local and global inverse covariances. Very few studies have addressed estimation of the local states. In fact, distributed inertial sensor systems consisting of several IMUs mounted in an aircraft affords both redundant inertial measurement information and distributed inertial state vectors, which can be used for aircraft navigation, guidance and control and also in the implementation of local motion compensation functions. These IMUs measure local motion with reference to specific coordinate frames defined by their installation positions and have individual error dynamics. Therefore, the local states must be accurately estimated to determine the local dynamic motion. The development of distributed filtering algorithms can also be used to investigate methods for dynamic alignment and calibration of distributed IMUs. To date, these considerations have not been addressed in the open literature.

4. CONCLUSIONS.

This paper has reviewed the developments of fault-tolerant aircraft navigation systems based on an extensive literature survey. The methodologies for the design and development of safety-critical aircraft navigation systems have been summarised, including fault-tolerant navigation system architectures, data fusion filter architectures and corresponding filtering algorithms. These methods provide the techniques for navigation system engineers and researchers to design and develop future aircraft multisensor navigation systems.

REFERENCES

[1] Spitzer, C. R. (2001). *The Avionics Handbook*. CRC Press.
[2] Tuttle, F. L., Kisslinger, R. L. and Ritzema, D. F. (1990). F-15 S/MTD IFPC fault tolerant design, *Proceedings of NAECON*, 501–506.

[3] McClary, C. R. (1992). A Fault-tolerant air data/inertial reference system. *IEEE Aerospace and Electronic Systems Magazine*, Vol. 7, No. 5, 19–23.

[4] Gertler, J. J. (1988). Survey of model-based failure detection and isolation in complex plants. *IEEE Control System Magazine*, *Vol. 8 No. 6*, 3–11.

[5] Patton, R. J. (1991). Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing & Control Engineering Journal*, *Vol. 2, No. 3*, 127–136.

[6] Hegg, J. (2002). Enhanced space integrated GPS/INS (SIGI). *IEEE Aerospace and Electronics Systems Magazine*, *Vol. 17, No. 4 April 2002, 26–33*.

[7] Ebner, R. E. and Klein, A. D. (1995). Integrated navigation/flight control for future high performance aircraft. *ADARD-AG-314, Analysis, Design and Synthesis Methods for Guidance and Control Systems. V1-8*.

[8] Kelley, R. T., Carlson, N. A. and Berning, S. (1994). Integrated inertial network. *Proceedings of IEEE PLANS*, 439–446.

[9] Rounds, S. and Casey, J. (1987). A Fully integrated GPS/Doppler/inertial navigation system. *Proceedings of the Institute of Navigation Technical Meeting*.

[10] Schneider, A. M. and Maida, J. L. (1988). A Kalman filter for an integrated Doppler/GPS navigation system. *Proceedings of IEEE PLANS*, 408–415.

[11] Hyslop, G., Gerth, D. and Kraemer, J. (1990). GPS/INS integration on the standoff land attack missile (SLAM). *IEEE Aerospace and Electronics Systems Magazine*, *Vol. 5*, No.7, 29–34.

[12] Diesel, J. W. (1991). GPS/INS integration for civil aviation. *Proceedings of National Telesystems Conference*, 223–228.

[13] Bell, W. B., Gorre, R. G. and Cockrell, L. D. (1998). Cascading filtered DTS data into a loosely coupled GPS/INS system. *Proceedings of IEEE PLANS*, 586–593.

[14] Schlee, F. H., Toda, N. F., Islam M. A. and Standish, C. J., (1998). Use of an external cascade Kalman filter to improve the performance of a Global Positioning System (GPS) inertial navigator. *Proceedings of IEEE NAECON*, 345–350.

[15] Wade, M. and Grewal, M. S. (1988). Analysis of a cascaded INS calibration filter. *Proceedings of the IEEE*, 366–373.

[16] Karatsinides, S. E. (1994). Enhancing filter robustness in cascaded GPS-INS integrations. *IEEE Trans. On Aerospace and Electronic Systems*, *Vol. 30, No. 4 Oct., 1994*, 1001–1008.

[17] Carlson, N. A. (1988). Federated filter for fault-tolerant integrated systems. *Proceedings of IEEE PLANS*, 110–119.

[18] Felter, S. C. (1990). An overview of decentralized Kalman filter techniques. *Proceedings of IEEE Southern Tier Technical Conference*, 79–87.

[19] Carlson, N. A. (1990). Federated square root filter for decentralized parallel processors. *IEEE Trans. On Aerospace and Electronic Systems*, *Vol. 26 No. 3* 517–525.

[20] Jr. Lawrence, P. J. and Berarducci, M. P. (1994). Comparison of federated and centralized Kalman filters with fault detection considerations. *Proceedings of IEEE PLANS*, 703–710.

[21] Levy, L. J. (1996). Suboptimality of cascaded and federated Kalman filters. *Proceedings of the ION 52nd Annual Meeting, Navigational Technology for the 3rd Millennium*, 399–407.

[22] Tupysev, V. A. (2000). Federated Kalman filter via formation of relation equations in augmented state space. *Journal of Guidance, Control and Dynamics*, Vol. **23**, No. 3, 391–398.

[23] Carlson, N. A. (2002). Federated filter for distributed navigation and tracking applications. *Proceedings of the ION 58th Annual Meeting and the CIGTF 21st Guidance Test Symposium*, 340–353.

[24] Liggins, M. E. II, Chong, C. Y, Kadar, I., Alford, M. G., Vannicola, V. and Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE, Vol. 85, No. 1*, 95–107.

[25] Speyer, J. L. (1979). Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem. *IEEE Trans. On Automatic Control*, *Vol. 24, No. 2*, 266–269.

[26] Willsky, A. S., Bello, M. G., Castanon, D. A., Levy, B. C. and Verghese, G. C. (1982). Combing and updating of local estimates and regional maps along sets of one-dimensional tracks. *IEEE Trans. On Automatic Control*, *Vol. 27, No. 4*, 799–813.

[27] Kerr, T. H. (1987). Decentralized filtering and redundancy management for multisensor navigation. *IEEE Trans. on Aerospace and Electronics Systems*, *Vol. AES-23, No. 1*, 83–119.

[28] Brumback, B. D. and Srinath, M. D. (1987). A Fault-tolerant multisensor navigation system design. *IEEE Trans. On Aerospace and Electronic Systems*, *Vol. 23, No. 6*, 738–755.

[29] Hashemipour, H. R., Roy, S. and Laub, A. J. (1988). Decentralized structure for parallel Kalman filtering. *IEEE Trans. on Automatic Control*, *Vol. 33, No. 1*, 88–94.

[30] Hong, L. (1991). Centralized and distributed multisensor integration with uncertainties in communication networks. *IEEE Trans. on Aerospace and Electronic Systems*, *Vol. 27, No. 2*, 370–379.

[31] Roy, S., Hashemi, R. H. and Laub, A. J. (1991). Square root parallel Kalman filtering using reduced-order local filter. *IEEE Trans. on Aerospace and Electronic Systems*, *Vol. 21, No. 2*, 276–289.

[32] Paik, B. S. and Oh, J. H. (2000). Gain fusion algorithm for decentralised parallel Kalman filters. *IEE Proceedings of Control Theory and Applications*, *Vol. 147, No. 1*, 97–103.

[33] Raol, J. R. and Girija, G. (2002). Sensor data fusion algorithms using square-root information filtering. *IEE Proceedings of Radar, Sonar and Navigation*, *Vol. 149, No. 2*, 89–96.

[34] Rao, B. S. and Durrant-Whyte, H. F.(1991). Fully decentralised algorithm for multisensor Kalman filtering. *IEE Proceedings-Control Theory and Applications*, *Vol. 138, No. 5*, 413–420.

[35] Berg, T. M. and Durrant-Whyte, H. F. (1992). Distributed and decentralized estimation. *Proceedings of international Conference on Intelligent Control and Instrumentation, SICICI'92*, *Vol. 2*. Singapore 1992, 1118–1123.