IET Radar Sonar Navigation., 2008, 2, (1), pp. 51–62. ISSN 1751-8784

Distributed data fusion algorithms for inertial network systems

D.J. Allerton and H. Jia

Abstract: New approaches to the development of data fusion algorithms for inertial network systems are described. The aim of this development is to increase the accuracy of estimates of inertial state vectors in all the network nodes, including the navigation states, and also to improve the fault tolerance of inertial network systems. An analysis of distributed inertial sensing models is presented and new distributed data fusion algorithms are developed for inertial network systems. The distributed data fusion algorithm comprises two steps: inertial measurement fusion and state fusion. The inertial measurement fusion allows each node to assimilate all the inertial measurements from an inertial network system, which can improve the performance of inertial sensor failure detection and isolation algorithms by providing more information. The state fusion further increases the accuracy and enhances the integrity of the local inertial states and navigation state estimates. The simulation results show that the two-step fusion procedure overcomes the disadvantages of traditional inertial sensor alignment procedures. The slave inertial nodes can be accurately aligned to the master node.

1 Introduction

The concept of an inertial network system in aircraft avionics was initially proposed by Kelley *et al.* [1] and subsequently developed by Berning *et al.* [2] and Kaiser *et al.* [3]. In this architecture, inertial sensor systems are located at several places in an aircraft in order to meet the fault tolerance requirements of aircraft navigation and to provide accurate local inertial state vectors for several airborne avionics systems. For example, fire/weapon control systems and radar search/tracking systems require accurate local and centre of gravity (cg)-referenced inertial state information to stabilise these systems and to compensate for local motion.

The development of inertial network systems arise from reliability requirements and increased reliance on inertial information, particularly in advanced military aircraft where airframe flexibility must be taken into account in the installation and alignment of airborne avionics systems. Inertial sensor systems provide flight-critical information for all safety or mission-critical avionic systems, such as flight control and navigation systems, as well as other airborne systems. Redundant inertial systems are used to provide the level of fault tolerance necessary in aircraft navigation systems, in order to meet safety and reliability requirements for civil or military aircraft. Typically, a combat platform may have 12 inertial measurement units (IMUs) of various quality providing the inertial state vector information needed in mission-critical avionics systems and weapon systems [2]. These IMUs are installed at different locations in a flexible airframe and both structural and in-flight misalignments between these sensors/weapon locations need to be estimated in order to align the dynamic sensors and weapon systems. For example, the performance of sensors/systems such as SAR and terrain following radar, pointing systems, forward-looking infrared receivers, laser spot trackers and missile pylons all depend on the precise alignment of the system sensitive axes in inertial networks. Fusion of measurements from distributed IMUs can provide highly reliable inertial vector information and can also be used to detect sensor failures and to reconfigure inertial measurements in inertial networks.

During the last few years, distributed and integrated modular avionics architectures have been introduced into modern aircraft systems [4, 5] as a result of advances in high-speed airborne data communication networks and embedded computer systems. These enabling technologies provide airborne avionics systems with powerful data processing and communication capability. Furthermore, the reducing cost, size and mass of sensors, including fibre optic gyros and microelectromechanical systems (MEMS) inertial sensor systems [6, 7], have enabled redundant inertial sensors to be integrated into a single IMU box. In these systems, non-orthogonal configurations can be used to improve system reliability and to reduce the cost, size and mass of aircraft navigation systems. In a non-orthogonal configured IMU, redundant inertial sensors are skewed with respect to the orthogonal frame; this configuration is known as a skewed redundant IMU (SRIMU).

Few researchers to date have focused on a systematic approach to the design of inertial network algorithms. This paper proposes innovative data fusion methods for inertial network systems to provide dynamic alignment and calibration of distributed inertial systems.

2 Inertial network system architectures

An inertial network architecture is illustrated in Fig. 1 in which each node represents an individual sensing location

© The Institution of Engineering and Technology 2008

doi:10.1049/iet-rsn:20060159

Paper first received 23rd November 2006 and in revised form 7th August 2007

D.J. Allerton is with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

H. Jia is with the Department of Aerospace Engineering, Cranfield University, Cranfield, Bedfordshire MK43 0AL, UK

E-mail: d.j.allerton@sheffield.ac.uk

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

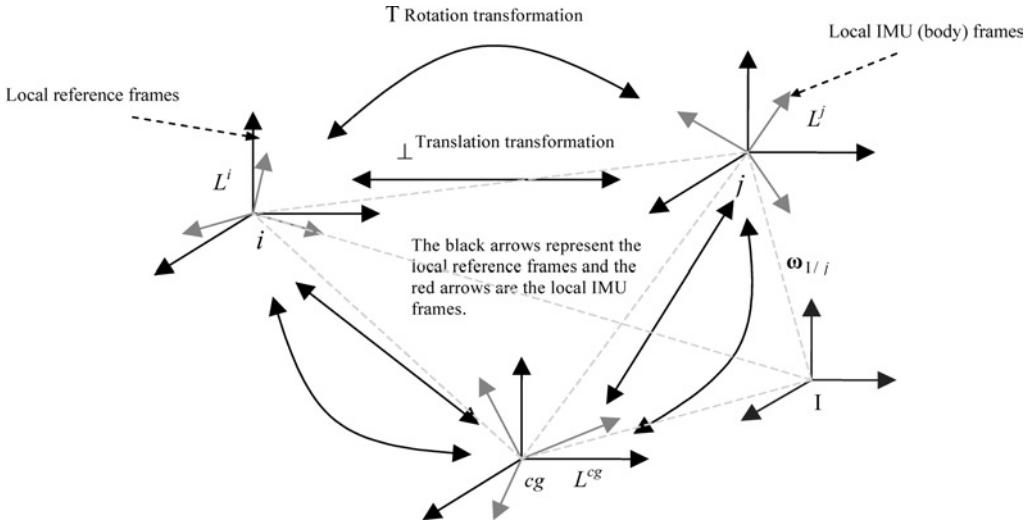

Fig. 1 *Relationships between IMU nodes*

consisting of an IMU and an embedded microprocessor. Each node is assumed to be able to communicate with other nodes, so that information from one node can be shared by other nodes. The IMU at each node can be integrated with other navigation-aiding systems, including global navigation satellite systems (GNSS), Doppler radar and other radio navigation systems. Avionics data buses are used to interconnect the IMU nodes to implement an inertial network system. The node located at the aircraft cg is usually a master node, also referred to as the cg node; the other nodes are the local nodes, known as slave nodes. The data fusion filter located at the cg node provides the navigation states and the cg-referenced inertial state vector; the data fusion filters located at the slave nodes provide the local states and the local inertial vector information.

This form of distributed inertial network system [1] is extended in this paper and affords the following advantages.

1. Fault tolerance and robustness to sensor/system failures. Data fusion algorithms are designed so that the failure of any node or element of the node will not lead to degradation of the performance of the aircraft navigation system. Moreover, degradation of the performance of local systems at a failed node will be gradual.
2. Flexibility. It is straightforward to add or remove one or more sensor systems in a distributed system network.
3. Highly reliable cg state estimation. The cg data fusion filter combines all local estimates with its own estimate to obtain the aircraft cg motion states, which are used to support aircraft navigation, flight control and guidance and other functions that require cg-referenced data.
4. Accurate local state estimation. The slave data fusion filter located at each node fuses all measurements from healthy inertial sensor systems to derive globally optimal estimates of the local states; these are used to support the stabilisation of various avionics system platforms and to provide local motion compensation.
5. Automatic alignment of low-quality sensors. Because information is shared by all nodes, distributed data fusion filters can use the local estimates obtained at a node with a high-quality IMU, to dynamically correct and align low-quality IMUs situated at other nodes. Consequently, traditional inertial system alignment algorithms, which may necessitate specific manoeuvers to be flown, are no longer necessary in distributed inertial network systems.

3 Distributed inertial sensing models

3.1 Distributed node frames

Although IMUs in an inertial network independently measure individual local qualities, the measured or estimated states are not completely independent; they are dynamically related to each other owing to the rigid structure of the aircraft. The development of this dynamic relationship between the local measured states enables the inertial information provided by an inertial network to be used to detect and isolate sensor/system failures and to implement the dynamic alignments between different inertial systems.

Fig. 1 illustrates three local IMU frames and the corresponding local reference frames. Let I be an inertial reference frame, let cg be the local body frame of the IMU node located at the aircraft centre of gravity i and j represent the slave IMU nodes and their individual local body frames. In this paper, the IMU frames are assumed to be aligned with the local body frames when the IMUs are installed at local locations in the aircraft. $\perp$ represents a translational transformation, for example, the translation vector from nodes i to j is denoted by $\perp^j_i$. $\boldsymbol{T}$ represents a rotational transformation, for example, the transformation matrix from nodes i to j is denoted by $\boldsymbol{T}^j_i$. Exchanging the superscript and the subscript of a transformation matrix represents an inverse of this transformation, for example, $(\boldsymbol{T}^j_i)^{-1} = \boldsymbol{T}^i_j$. Let the local reference frames at the nodes i, j and cg be denoted by L^i, L^j and L^{cg}, respectively, then the corresponding transformations from the local reference frames to the local body frames are given by $\boldsymbol{T}^i_{L^i}$, $\boldsymbol{T}^j_{L^j}$ and $\boldsymbol{T}^{cg}_{L^{cg}}$. If the local level frames, such as the north-east-down, are used as the local reference frames, these rotation matrices represent the orientations of the local body axes relative to the local level frames. Because the origins of the local level frames are defined with respect to the geodetic coordinates of the IMU nodes and the magnitudes of the translation vectors between these IMU nodes are very small, the misalignments between the local level frames, caused by the translation vectors, can be ignored. In order to simplify the development of an inertial network sensing model, it is assumed that the local-level frames located at all the inertial network nodes are identical, that is $L^i = L^j = L^{cg}$.

Let the relative rotation of the IMU frame i with respect to the frame j be $\boldsymbol{\omega}_{j/i}$ and its inverse rotation be $\boldsymbol{\omega}_{i/j} = -\boldsymbol{\omega}_{j/i}$. From multi-body rotation mechanics, the

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

absolute angular velocity of an IMU frame in an inertial network system is the sum of the absolute angular velocity of another IMU frame and the relative angular velocity between these two frames. Therefore, the absolute angular rate vectors measured by the local IMUs can be represented as follows

$$\boldsymbol{\omega}_{I/i} = \boldsymbol{\omega}_{I/\mathrm{cg}} + \boldsymbol{\omega}_{\mathrm{cg}/i} \tag{1}$$

$$\boldsymbol{\omega}_{I/j} = \boldsymbol{\omega}_{I/\mathrm{cg}} + \boldsymbol{\omega}_{\mathrm{cg}/j} \tag{2}$$

$$\boldsymbol{\omega}_{I/\mathrm{cg}} = \boldsymbol{\omega}_{I/j} + \boldsymbol{\omega}_{j/\mathrm{cg}} \tag{3}$$

The rotational transformations between the local IMU frames depend on the relative angular velocities between these frames. In the stationary (rigid airframe) case, there is no relative angular motion between the IMU nodes; in the dynamic case, the relative angular rates between different IMU frames can be treated as a random variable.

3.2 Stationary inertial sensing model

In this model, the aircraft structure is assumed to be rigid. The dynamic relationships between different IMU frames can be described by fixed rotational and translational transformations, which can be precisely determined from the geometry of IMU locations at the time of installation or subsequently estimated using a distributed alignment Kalman filter.

If the local state $\boldsymbol{x}$ is a rate vector (such as acceleration, velocity or angular velocity), a rotation matrix is used to complete the rotation transformation between node frames, for example, from nodes j to i, as follows

$$\boldsymbol{x}_i = \boldsymbol{T}_j^i \boldsymbol{x}_j \tag{4}$$

If the local state $\boldsymbol{x}$ is a displacement vector, a rotation matrix is combined with a translation vector to represent the transformation between frames as follows

$$\boldsymbol{x}_i = \boldsymbol{T}_j^i \boldsymbol{x}_j + \perp_j^i \tag{5}$$

where the states $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are expressed in their individual local frames, $\boldsymbol{T}_j^i$ a known rotation matrix and $\perp_j^i$ a known translation vector expressed relative to node i.

For the case where the local states are the Euler angles, the attitude matrix transformation from one node to the other has the following form

$$\boldsymbol{T}_{L^i}^i = \boldsymbol{T}_j^i \boldsymbol{T}_{Lj}^j \tag{6}$$

It should be noted that the new Euler angles in (6) are computed from the matrix $\boldsymbol{T}_{L^i}^i$.

If a node IMU is an SRIMU, the measured inertial states, accelerations and angular rates, are expressed in terms of the local body frame but the IMU outputs are represented in the inertial instrument frame, which defines the sensing directions of inertial sensors and differs from the IMU frame. The transformation between the inertial instrument frame and the local IMU frame, for example, at the cg node, is given by $\boldsymbol{H}_{\mathrm{imu}^{\mathrm{cg}}}^{\mathrm{cg}}$ where the subscript $\mathrm{imu}^{\mathrm{cg}}$ represents the inertial instrument frame. $\boldsymbol{H}_{\mathrm{imu}^{\mathrm{cg}}}^{\mathrm{cg}}$ (referred to as the design matrix) depends on the SRIMU configuration and can be dynamically reconfigured [8, 9]. The SRIMU measurement vector at the cg node, $\boldsymbol{m}_{\mathrm{imu}^{\mathrm{cg}}}$, can be rewritten as follows

$$\boldsymbol{m}_{\mathrm{imu}^{\mathrm{cg}}} = \boldsymbol{H}_{\mathrm{cg}}^{\mathrm{imu}^{\mathrm{cg}}} \boldsymbol{x}_{\mathrm{cg}} \tag{7}$$

By applying rotation transformations, the measurement vectors of SRIMUs located at nodes i and j, $\boldsymbol{m}_{\mathrm{imu}^i}$ and $\boldsymbol{m}_{\mathrm{imu}^j}$, can be represented in terms of the local body frame at the cg node as follows

$$\boldsymbol{m}_{\mathrm{imu}^i} = \boldsymbol{H}_i^{\mathrm{imu}^i} \boldsymbol{x}_i = \boldsymbol{H}_i^{\mathrm{imu}^i} \boldsymbol{T}_{\mathrm{cg}}^i \boldsymbol{x}_{\mathrm{cg}} \tag{8}$$

$$\boldsymbol{m}_{\mathrm{imu}^j} = \boldsymbol{H}_j^{\mathrm{imu}^j} \boldsymbol{x}_j = \boldsymbol{H}_j^{\mathrm{imu}^j} \boldsymbol{T}_{\mathrm{cg}}^j \boldsymbol{x}_{\mathrm{cg}} \tag{9}$$

where $\boldsymbol{H}_i^{\mathrm{imu}^i}$ and $\boldsymbol{H}_j^{\mathrm{imu}^j}$ are the SRIMU design matrices at nodes i and j, respectively.

From (8) and (9), the cg node assimilates the inertial measurement information from the slave nodes i and j. Therefore the total inertial measurement at the cg node is represented as

$$\boldsymbol{m}_{\mathrm{cg}} \equiv \begin{bmatrix} \boldsymbol{m}_{\mathrm{imu}^{\mathrm{cg}}} \\ \boldsymbol{m}_{\mathrm{imu}^i} \\ \boldsymbol{m}_{\mathrm{imu}^j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{\mathrm{cg}}^{\mathrm{imu}^{\mathrm{cg}}} \\ \boldsymbol{H}_i^{\mathrm{imu}^i} \boldsymbol{T}_{\mathrm{cg}}^i \\ \boldsymbol{H}_j^{\mathrm{imu}^j} \boldsymbol{T}_{\mathrm{cg}}^j \end{bmatrix} \boldsymbol{x}_{\mathrm{cg}} = \boldsymbol{H}_{\mathrm{cg}} \boldsymbol{x}_{\mathrm{cg}} \tag{10}$$

Similarly, the inertial measurement vectors at nodes i and j are as follows

$$\boldsymbol{m}_i \equiv \begin{bmatrix} \boldsymbol{m}_{\mathrm{imu}^{\mathrm{cg}}} \\ \boldsymbol{m}_{\mathrm{imu}^i} \\ \boldsymbol{m}_{\mathrm{imu}^j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{\mathrm{cg}}^{\mathrm{imu}^{\mathrm{cg}}} \boldsymbol{T}_i^{\mathrm{cg}} \\ \boldsymbol{H}_i^{\mathrm{imu}^i} \\ \boldsymbol{H}_j^{\mathrm{imu}^j} \boldsymbol{T}_i^j \end{bmatrix} \boldsymbol{x}_i = \boldsymbol{H}_i \boldsymbol{x}_i \tag{11}$$

$$\boldsymbol{m}_j \equiv \begin{bmatrix} \boldsymbol{m}_{\mathrm{imu}^{\mathrm{cg}}} \\ \boldsymbol{m}_{\mathrm{imu}^i} \\ \boldsymbol{m}_{\mathrm{imu}^j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{\mathrm{cg}}^{\mathrm{imu}^{\mathrm{cg}}} \boldsymbol{T}_j^{\mathrm{cg}} \\ \boldsymbol{H}_i^{\mathrm{imu}^i} \boldsymbol{T}_j^i \\ \boldsymbol{H}_j^{\mathrm{imu}^j} \end{bmatrix} \boldsymbol{x}_j = \boldsymbol{H}_j \boldsymbol{x}_j \tag{12}$$

Equations (10)–(12) indicate that each node shares the same redundant inertial measurements even though all the IMUs are traditional orthogonal systems. Various weighted least-squares estimators can be applied to the redundant measurement equations to estimate the local inertial states. This data assimilation and weighted least-squares estimation is referred to as inertial measurement (data) fusion in this paper. Furthermore, many failure detection and isolation algorithms, such as the parity space-based methods and generalised likelihood ratio test algorithms described in [8, 10, 11], can be applied to these equations to detect and isolate inertial sensor failures in the inertial network system. As a result, the use of inertial measurement fusion procedures increases the accuracy of the local inertial state estimates at each IMU node and improves the performance of the navigation system.

From (6), one IMU node can also assimilate attitude information from other IMU nodes using the following transformations. At node j, the local attitude matrix is

$$\boldsymbol{T}_{L^i}^i \tag{13}$$

The assimilated attitude matrices are

$$\boldsymbol{T}_{Lj}^i = \boldsymbol{T}_j^i \boldsymbol{T}_{Lj}^j \tag{14}$$

$$\boldsymbol{T}_{L^{\mathrm{cg}}}^i = \boldsymbol{T}_{\mathrm{cg}}^i \boldsymbol{T}_{L^{\mathrm{cg}}}^{\mathrm{cg}} \tag{15}$$

As a result, the redundant attitude information at node i consists of (13)–(15). At node j, the local attitude matrix is

$$\boldsymbol{T}_{Lj}^j \tag{16}$$

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

The assimilated attitude matrices are

$$\boldsymbol{T}^{j}_{L^i} = \boldsymbol{T}^{j}_{i}\boldsymbol{T}^{i}_{L^i} \tag{17}$$

$$\boldsymbol{T}^{j}_{L^{\mathrm{cg}}} = \boldsymbol{T}^{j}_{\mathrm{cg}}\boldsymbol{T}^{\mathrm{cg}}_{L^{\mathrm{cg}}} \tag{18}$$

Similarly, the redundant attitude information at node j consists of (16)–(18).

At node cg, the local attitude matrix is

$$\boldsymbol{T}^{\mathrm{cg}}_{L^{\mathrm{cg}}} \tag{19}$$

The assimilated attitude matrices are

$$\boldsymbol{T}^{\mathrm{cg}}_{L^i} = \boldsymbol{T}^{\mathrm{cg}}_{i}\boldsymbol{T}^{i}_{L^i} \tag{20}$$

$$\boldsymbol{T}^{\mathrm{cg}}_{Lj} = \boldsymbol{T}^{\mathrm{cg}}_{j}\boldsymbol{T}^{j}_{Lj} \tag{21}$$

where the redundant attitude information at node cg consists of (19)–(21).

The redundant attitude matrix information at each node can be fused by a weighted least-squares estimator to increase the accuracy of the local attitude estimates and to improve the fault tolerance of the navigation states.

3.3 Dynamic inertial sensing model

Although the assumption of a rigid body aircraft applies to a wide range of applications in aircraft navigation and control systems, this assumption may be invalid in many military applications where precise local inertial states are needed including targeting systems, pointing systems and other weapon systems. During high-speed flight and high-dynamic manoeuvres, the airframe should be considered as a flexible structure. The rotational matrices given in the previous section are no longer stationary but are time-varying dynamic rotation matrices. If the flexible structure of an aircraft is ignored, these matrices introduce errors in the rotation transformation, leading to large errors in the estimates of the local states. Accordingly, it is necessary to develop the dynamic relationships between the network nodes and to estimate these dynamic transformation matrices in flight. Carison *et al.* [12] suggest a differential inertial filter (DIF) to estimate the angular flexing of the slave IMU frames with respect to a reference IMU frame, usually the cg IMU node frame. The DIF method processes a differential or delta inertial state vector, which is the difference between the cg and slave IMU measurements. However, this method depends on two vital conditions. First, compensation is needed for the lever-arm accelerations and flexing angular rates before initiating the DIF. For significant flexing motion of an airframe, this compensation is difficult to compute, particularly during manoeuvres because the flexing of the slave IMU node frames (with respect to the cg frame) changes with flight conditions. Secondly, the DIF dynamic model is particularly sensitive to aircraft manoeuvres and the resultant flexing motion. Two methods are presented in this paper to determine these dynamic rotational transformations, which avoid compensation for lever-arm accelerations and flexing angular rates between the cg and the slave IMU nodes. The first approach is an iterative estimation method and the second method establishes analytical models of the rotation matrices. In both methods, it is assumed that the initial transformation matrices are known. In practice, these matrices can be determined from the stationary transformation matrices described in the previous section, when the aircraft is on the ground or in level non-accelerating flight.

3.3.1 Iterative estimating method: The iterative estimation method is based on (6) where the local attitude matrices are computed at all the IMU nodes by invoking the inertial attitude determination algorithm. The dynamic transformation matrices are then estimated from the computed local attitude matrices.

Because the dynamic change of the rotation matrix relative to its initial matrix is generally within a small dynamic range, the current estimate of the rotation transformation $\boldsymbol{T}^{j}_{ik}$ can be approximated by the combination of the previous estimate and a small angle displacement vector $\boldsymbol{\psi}^{j}_{ik}$. The estimated rotation matrix at the current time can be rewritten as

$$\hat{\boldsymbol{T}}^{j}_{ik} \approx \hat{\boldsymbol{T}}^{j}_{ik-1}(\boldsymbol{I} + \boldsymbol{\psi}^{j}_{i_k} \times) = \hat{\boldsymbol{T}}^{j}_{Lj}\hat{\boldsymbol{T}}^{L^i}_{i}$$

where the subscript k represents the iteration step.

Therefore the iterative computation equation can be given as follows

$$(\boldsymbol{I} + \boldsymbol{\psi}^{j}_{ik} \times) = [\hat{\boldsymbol{T}}^{j}_{ik-1}]^{-1}\hat{\boldsymbol{T}}^{j}_{Lj}\hat{\boldsymbol{T}}^{L^i}_{i} \tag{22}$$

The method is outlined in Fig. 2. When the norm of $\boldsymbol{\psi}^{j}_{ik}$ is less than a specified threshold value, the iterative process terminates and the current transformation matrix can be determined.

Although an iterative technique can be a time-consuming procedure, because the inertial attitude determination algorithms may be repeated for several times at all the IMU

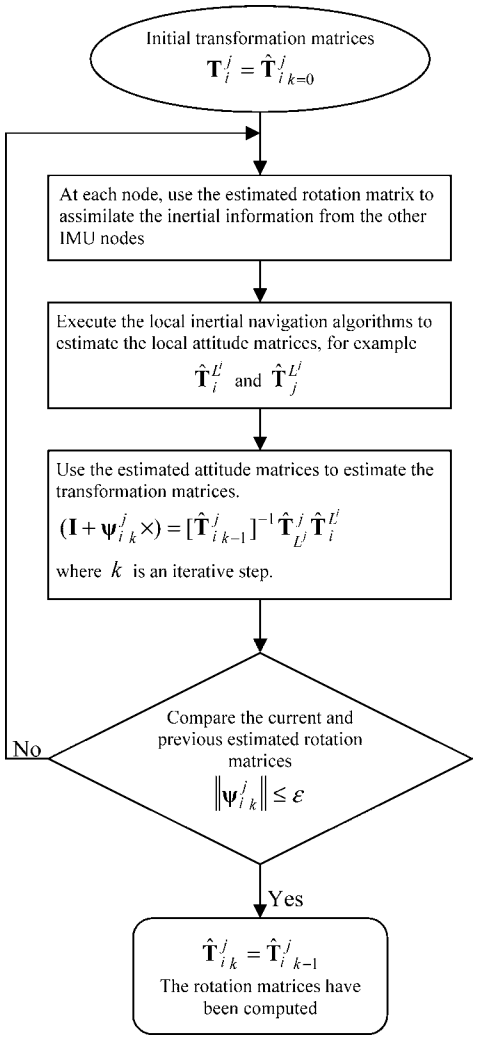

Fig. 2 *Iterative computation of rotation matrices*

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

nodes at each iteration, the advantage of this method is that the errors of the estimates of the rotation matrices are independent of the dynamic models of the rotation matrices. Uncertainties in the local attitude matrix estimates can contribute to errors in the estimation of the rotation matrix. However, the inertial attitude determination algorithms provide an effective filter, which can reduce the effect of IMU measurement noise on the rotation matrix estimates.

3.3.2 Analytical method: This method is based on the development of analytical dynamic models of the transformation matrices. If the cg body frame is used as a reference frame to represent the relative rotation motion of the other frames and the measured angular velocities, (1)–(3) can be rewritten in terms of angular rates as follows

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/i} = \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}} + \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{cg}/i} \tag{23}$$

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/j} = \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}} + \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{cg}/j} \tag{24}$$

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/i} = \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/j} + \boldsymbol{\Omega}^{\mathrm{cg}}_{j/i} \tag{25}$$

where $\boldsymbol{\Omega}$ is a skew-symmetric matrix of the corresponding angular rate vector $\boldsymbol{\omega}$. The superscript cg denotes that the angular rate vectors are expressed in terms of the cg body coordinates.

In terms of attitude matrix differential equations, (23) can be written as

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/i} = \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}} - \hat{\boldsymbol{T}}^{i}_{\mathrm{cg}}\boldsymbol{T}^{\mathrm{cg}}_{i} \tag{26}$$

Therefore

$$\hat{\boldsymbol{T}}^{i}_{\mathrm{cg}} = (\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}} - \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/i})\boldsymbol{T}^{i}_{\mathrm{cg}} \tag{27}$$

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/i} = \boldsymbol{T}^{\mathrm{cg}}_{i}\boldsymbol{\Omega}^{i}_{\mathrm{l}/i}\boldsymbol{T}^{i}_{\mathrm{cg}} \tag{28}$$

Similarly, the differential equation of the rotation matrix $\boldsymbol{T}^{j}_{\mathrm{cg}}$ is given by

$$\hat{\boldsymbol{T}}^{j}_{\mathrm{cg}} = (\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}} - \boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/j})\boldsymbol{T}^{j}_{\mathrm{cg}} \tag{29}$$

$$\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l}/j} = \boldsymbol{T}^{\mathrm{cg}}_{j}\boldsymbol{\Omega}^{j}_{\mathrm{l}/j}\boldsymbol{T}^{j}_{\mathrm{cg}} \tag{30}$$

where $\boldsymbol{\Omega}^{\mathrm{cg}}_{\mathrm{l/cg}}$, $\boldsymbol{\Omega}^{i}_{\mathrm{l}/i}$ and $\boldsymbol{\Omega}^{j}_{\mathrm{l}/j}$ consist of the local absolute angular rate vectors, estimated from the local IMU measurements at nodes cg, i and j.

The rotational transformation matrix between j and i is then computed from the following equation

$$\boldsymbol{T}^{i}_{j} = \boldsymbol{T}^{i}_{\mathrm{cg}}\boldsymbol{T}^{\mathrm{cg}}_{j} \tag{31}$$

Clearly, the dynamic models of the rotation matrices are non-linear matrix differential equations, where the initial matrices can be derived from the stationary transformations and these differential equations are solved at each measurement time.

In comparison with the iterative method, the analytical method avoids the iterative computation of the inertial attitude determination algorithms at all the IMU nodes. However, the IMU measurement noise may affect the accuracy of the solutions of the rotation matrices because the IMU outputs are directly used in the rotation matrix differential equations. Consequently, pre-processing filters may be required to reduce the measurement noise. If the node IMUs have a good quality (say better than $0.05^{\circ}/\mathrm{h}$), the iterative method is especially effective in determination of the rotation matrices between nodes. The analytical method is more applicable to dynamic IMU alignment in an inertial network system.

4 Distributed data fusion algorithms

4.1 Kalman filter algorithm

The distributed inertial network fusion algorithms developed in this paper are based on Kalman filtering techniques. Consider the discrete-time stochastic process where the system and measurement models are given by

$$\boldsymbol{x}(t_k) = \Phi(t_k, t_{k-1})\boldsymbol{x}(t_{k-1}) + \boldsymbol{G}(t_{k-1})\boldsymbol{w}(t_{k-1}) \tag{32}$$

$$\boldsymbol{z}(t_k) = \boldsymbol{D}(t_k)\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k) \tag{33}$$

where Φ is an $n \times n$ state transition matrix, $\boldsymbol{D}$ an $m \times n$ measurement matrix, $\boldsymbol{x}(t_k)$ an n-system state, $\boldsymbol{w}(t_k)$ a q-additive process noise, which takes into account the perturbations to the system, $\boldsymbol{G}(t_k)$ an $n \times q$ matrix, $\boldsymbol{z}(t_k)$ an m-measurement vector and $\boldsymbol{v}(t)$ is an m-additive measurement noise vector. It is assumed that the noise vectors $\boldsymbol{w}(t_k)$ and $\boldsymbol{v}(t)$ are independent, zero-mean, white Gaussian sequences of covariance $\boldsymbol{Q}(t_k)$ and $\boldsymbol{R}(t_k)$, respectively. The initial system state $\boldsymbol{x}(t_0)$ is a Gaussian distributed random variable and is independent of the noise, with an initial value $\boldsymbol{x}_0$ and covariance $\boldsymbol{P}_0$.

The Kalman filter algorithm is as follows [13].

Step 1: Initialization

$$\boldsymbol{P}(t_0) = \boldsymbol{P}_0;\ \hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0$$

Step 2: Time update (effect of dynamics, predictor)

$$\hat{\boldsymbol{x}}(t_k^-) = \Phi(t_k, t_{k-1})\hat{\boldsymbol{x}}(t_{k-1}^+)$$

$$\boldsymbol{P}(t_k^-) = \Phi(t_k, t_{k-1})\boldsymbol{P}(t_{k-1}^+)\Phi^{\mathrm{T}}(t_k, t_{k-1}) + \boldsymbol{G}(t_{k-1})\boldsymbol{Q}(t_{k-1})\boldsymbol{G}^{\mathrm{T}}(t_{k-1})$$

$$\boldsymbol{r}(t_k) = \boldsymbol{z}(t_k) - \boldsymbol{D}(t_k)\hat{\boldsymbol{x}}(t_k^-)$$

$$\boldsymbol{S}(t_k) = \boldsymbol{D}(t_k)\boldsymbol{P}(t_k^-)\boldsymbol{D}^{\mathrm{T}}(t_k) + \boldsymbol{R}(t_k)$$

Step 3: Measurement update (effect of measurement, estimator)

$$\boldsymbol{K}(t_k) = \boldsymbol{P}(t_k^-)\boldsymbol{D}^{\mathrm{T}}(t_k)\boldsymbol{S}^{-1}(t_k)$$

$$\hat{\boldsymbol{x}}(t_k^+) = \hat{\boldsymbol{x}}(t_k^-) + \boldsymbol{K}(t_k)\boldsymbol{r}(t_k)$$

$$\boldsymbol{P}(t_k^+) = \boldsymbol{P}(t_k^-) - \boldsymbol{K}(t_k)\boldsymbol{D}(t_k)\boldsymbol{P}(t_k^-)$$

From the predictor and the estimator equations given above, the Kalman filter outputs valuable statistical information that can be used to monitor both the convergence and the consistency of the filter estimation procedure. The outputs of the predictor include the filter innovation $\boldsymbol{r}(t_k)$ and its covariance $\boldsymbol{S}(t_k)$, whereas the estimator outputs the filter residual $\boldsymbol{r}(t_k^+)$ and the residual covariance $\boldsymbol{S}(t_k^+)$, defined as follows

$$\boldsymbol{r}(t_k^+) = \boldsymbol{z}(t_k) - \boldsymbol{D}(t_k)\boldsymbol{x}(t_k^+)$$

$$\boldsymbol{S}(t_k^+) = \boldsymbol{H}(t_k)\boldsymbol{P}(t_k^+)\boldsymbol{D}^{\mathrm{T}}(t_k) + \boldsymbol{R}(t_k)$$

It has been shown that the filter innovation and residual processes are a zero-mean white Gaussian random sequence

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

in normal operation if the Kalman filter model matches the true system model [13, 14]. This feature can be exploited in the analysis of the Kalman filter integrity to check the consistency of measurement data for sensor failure detection and to monitor the filter divergence.

4.2 Distributed inertial data fusion algorithm

Assume that all the local IMUs are independent of each other and their measurements have a Gaussian probability distribution. Then, the errors of the local inertial state estimates are also a Gaussian distributed random vector and the probability density function of the local inertial state is given by

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^3 \det \boldsymbol{P}_x}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{P}_x^{-1}(\boldsymbol{x}-\hat{\boldsymbol{x}})\right] \quad (34)$$

where $\boldsymbol{x}$ is a three-dimensional local inertial state vector, for example, an acceleration or angular rate vector, and $\boldsymbol{P}_x$ is the covariance matrix of the error of the local inertial state estimate. From least-squares estimation

$$\boldsymbol{P}_x = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1} = \boldsymbol{H}^{*}\boldsymbol{R}[\boldsymbol{H}^{*}]^{\mathrm{T}} \quad (35)$$

where $\boldsymbol{H}^{*} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}$ is the pseudo-inverse matrix of the IMU design matrix $\boldsymbol{H}$ (the superscripts and subscripts of the design matrix $\boldsymbol{H}$ have been omitted to simplify the expression).

The objective of inertial measurement fusion is to generate optimal estimates of all the local inertial states. If the optimisation criterion is defined as the maximum of the conditional probability

$$P(\boldsymbol{x}|\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_{\mathrm{cg}})$$

then, because all the IMU measurements are independent, the conditional probability density function of the true local inertial state at each IMU node can be represented as follows

$$p(\boldsymbol{x}|\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_{\mathrm{cg}}) = p(\boldsymbol{x}) = p(\boldsymbol{x}|\hat{\boldsymbol{x}}_i)p(\boldsymbol{x}|\hat{\boldsymbol{x}}_j)p(\boldsymbol{x}|\hat{\boldsymbol{x}}_{\mathrm{cg}}) \quad (36)$$

Applying the maximum likelihood estimator to (36) and incorporating (10)–(12), the inertial data fusion algorithms at each IMU node can be derived as follows

$$\boldsymbol{x}_J = \left[\sum_{l=i,j,\mathrm{cg}} \boldsymbol{T}_l^{J}\boldsymbol{P}_{x,l}^{-1}\boldsymbol{T}_J^{l}\right]^{-1} \sum_{l=i,j,\mathrm{cg}} \boldsymbol{T}_l^{J}\boldsymbol{P}_{x,l}^{-1}\boldsymbol{H}_l^{*}\boldsymbol{m}_l,$$
$$J = i, j, \mathrm{cg} \quad (37)$$

or

$$\boldsymbol{x}_J = \left[\sum_{l=i,j,\mathrm{cg}} \boldsymbol{T}_l^{J}\boldsymbol{P}_{x,l}^{-1}\boldsymbol{T}_J^{l}\right]^{-1} \sum_{l=i,j,\mathrm{cg}} \boldsymbol{T}_l^{J}\boldsymbol{H}_l^{\mathrm{T}}\boldsymbol{R}_l^{-1}\boldsymbol{m}_l \quad (38)$$

$$\boldsymbol{P}_{x,J}^{-1} = \sum_{l=i,j,\mathrm{cg}} \boldsymbol{T}_l^{J}\boldsymbol{P}_{x,l}^{-1}\boldsymbol{T}_J^{l}, \quad J = i, j, \mathrm{cg} \quad (39)$$

Equations (37)–(39) comprise the inertial measurement fusion algorithm at each IMU node. Although inertial measurement fusion is mainly used to provide reliable and accurate local inertial state estimates, one additional benefit of this approach is that the outputs can be used to detect and isolate inertial sensor failures if deviations of the estimator residuals are used as a test statistic. In effect, the inertial measurement fusion is a pre-processing procedure for the second-stage fusion.

4.3 State fusion filtering algorithm

From (32) and (33), the local dynamic models embedded in each node of the inertial network system can be described as follows

$$\boldsymbol{x}_J(t_k) = \boldsymbol{\Phi}_J(t_k, t_{k-1})\boldsymbol{x}_J(t_{k-1}) + \boldsymbol{G}_J(t_{k-1})\boldsymbol{w}_J(t_{k-1}) \quad (40)$$
$$\boldsymbol{z}_J(t_k) = \boldsymbol{D}_J(t_k)\boldsymbol{x}_J(t_k) + \boldsymbol{v}_J(t_k) \quad (41)$$

where $J = i, j$, cg denotes the IMU nodes.

The normalised measurement models of navigation aiding systems, the normalised SRIMU error dynamic models and the normalised error dynamic models of inertial systems are given in [15–17]. These models can be applied to all the IMU nodes if the corresponding coordinate frames are specified. As different frames are used in the individual nodes, each local dynamic model describes its local states, which will be different from the local states represented by the other dynamic models. The local state vector $\boldsymbol{x}_J$ can be partitioned into the local system state $\boldsymbol{x}_{1J}$ and the local sensor error state $\boldsymbol{x}_{2J}$, that is

$$\boldsymbol{x}_J = [\,\boldsymbol{x}_{1J}^{\mathrm{T}} \quad \boldsymbol{x}_{2J}^{\mathrm{T}}\,]^{\mathrm{T}} \quad (42)$$

The local system states at the nodes are referred to as similar states and the transformations between these similar states are given by the dynamic transformation matrices.

The measurement vector $\boldsymbol{z}_J$ can be decomposed into three sub-vectors as follows

$$\boldsymbol{z}_J = [\,\boldsymbol{z}_{JL}^{\mathrm{T}} \quad \boldsymbol{z}_{JS}^{\mathrm{T}} \quad \boldsymbol{z}_{JA}^{\mathrm{T}}\,]^{\mathrm{T}} \quad (43)$$

where $\boldsymbol{z}_{JL}$ is the measurement vector provided by the local sensor systems, $\boldsymbol{z}_{JS}$ is the measurement vector provided by the navigation-aiding systems and $\boldsymbol{z}_{JA}$ is the combination of the inertial measurements assimilated from other IMU nodes. Because these three measurement vectors are independent of each other, the following decompositions can be obtained

$$\boldsymbol{D}_J = [\,\boldsymbol{D}_{JL}^{\mathrm{T}} \quad \boldsymbol{D}_{JS}^{\mathrm{T}} \quad \boldsymbol{D}_{JA}^{\mathrm{T}}\,]^{\mathrm{T}} \quad (44)$$
$$\boldsymbol{v}_J = [\,\boldsymbol{v}_{JL}^{\mathrm{T}} \quad \boldsymbol{v}_{JS}^{\mathrm{T}} \quad \boldsymbol{v}_{JA}^{\mathrm{T}}\,]^{\mathrm{T}} \quad (45)$$
$$\boldsymbol{R}_J = \mathrm{blockdiag}[\,\boldsymbol{R}_{JL} \quad \boldsymbol{R}_{JS} \quad \boldsymbol{R}_{JA}\,] \quad (46)$$

The architecture of the state fusion filter algorithm at each node is illustrated in Fig. 3 where the local Kalman

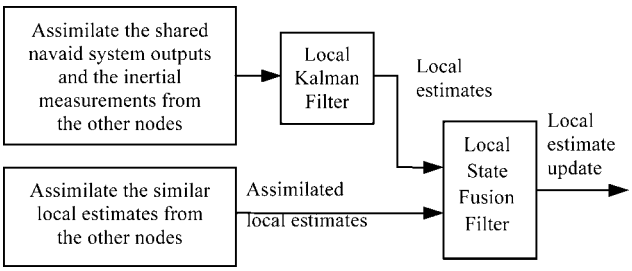

Fig. 3 *State fusion algorithm architecture*

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

filter uses the assimilated sensor measurements to estimate the local state. The local state fusion filter combines the local estimate and the assimilated estimates from the other nodes to update the local estimate.

All the local Kalman filters process the three forms of the measurements to obtain the local estimates. Applying the Kalman filtering algorithm to (40) and (41), and considering (43), (44) and (46) yields

$$\hat{\boldsymbol{x}}_J(t_k^-) = \boldsymbol{\Phi}_J(t_k, t_{k-1})\bar{\boldsymbol{x}}_J(t_{k-1}) \tag{47}$$

$$\begin{aligned}\boldsymbol{P}_J(t_k^-) &= \boldsymbol{\Phi}_J(t_k, t_{k-1})\bar{\boldsymbol{P}}_J(t_{k-1}^+)\boldsymbol{\Phi}_J^{\mathrm{T}}(t_k, t_{k-1}) \\ &\quad + \boldsymbol{G}_J(t_{k-1})\boldsymbol{Q}_J(t_{k-1})\boldsymbol{G}_J^{\mathrm{T}}(t_{k-1})\end{aligned} \tag{48}$$

$$\begin{aligned}\boldsymbol{P}_J^{-1}(t_k^+) &= \boldsymbol{P}_J^{-1}(t_k^-) + \boldsymbol{D}_J^{\mathrm{T}}(t_k)\boldsymbol{R}_J^{-1}(t_k)\boldsymbol{D}_J(t_k) \\ &= \boldsymbol{P}_J^{-1}(t_k^-) + \sum_{k=JL,JS,JA} \boldsymbol{D}_l^{\mathrm{T}}(t_k)\boldsymbol{R}_l^{-1}(t_k)\boldsymbol{D}_l(t_k)\end{aligned} \tag{49}$$

$$\begin{aligned}\boldsymbol{P}_J^{-1}(t_k^+)\hat{\boldsymbol{x}}_J(t_k^+) &= \boldsymbol{P}_J^{-1}(t_k^-)\hat{\boldsymbol{x}}_J(t_k^-) + \boldsymbol{D}_J^{\mathrm{T}}(t_k)\boldsymbol{R}_J^{-1}(t_k)\boldsymbol{z}_J(t_k) \\ &= \boldsymbol{P}_J^{-1}(t_k^-)\hat{\boldsymbol{x}}_J(t_k^-) + \sum_{l=JL,JS,JA} \boldsymbol{D}_l^{\mathrm{T}}(t_k)\boldsymbol{R}_l^{-1}(t_k)\boldsymbol{z}_l(t_k)\end{aligned} \tag{50}$$

where $\hat{\boldsymbol{x}}$ is the state estimated by the local Kalman filter and $\bar{\boldsymbol{x}}$ is the state updated by the local state fusion filter.

To update the locally estimated similar states at each node using the similar state estimates assimilated from the other nodes, a state fusion filter is needed in each node. Defining a quadratic cost function at node i as follows

$$F_i = \sum_{J=i,j,cg} (\boldsymbol{T}_J^i\boldsymbol{x}_J - \boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{T}_J^i\boldsymbol{P}_J^{-1}\boldsymbol{T}_i^J(\boldsymbol{T}_J^i\boldsymbol{x}_J - \boldsymbol{x}_i) \tag{51}$$

where $\boldsymbol{x}_i$ is the true local similar state at i and F_i is a cost function used to measure the displacement of the local state estimate from its true value.

The state fusion filter is designed to minimise F_i and is referred to as the minimum weighted mean square error criterion. Differentiating F_i and setting the result to zero yields

$$\bar{\boldsymbol{x}}_i = \bar{\boldsymbol{P}}_i^{-1}(\boldsymbol{P}_i^{-1}\hat{\boldsymbol{x}}_i + \boldsymbol{T}_j^i\boldsymbol{P}_j^{-1}\hat{\boldsymbol{x}}_j + \boldsymbol{T}_{\mathrm{cg}}^i\boldsymbol{P}_{\mathrm{cg}}^{-1}\hat{\boldsymbol{x}}_{\mathrm{cg}}) \tag{52}$$

$$\bar{\boldsymbol{P}}_i^{-1} = (\boldsymbol{P}_i^{-1} + \boldsymbol{T}_j^i\boldsymbol{P}_j^{-1}\boldsymbol{T}_i^j + \boldsymbol{T}_{\mathrm{cg}}^i\boldsymbol{P}_{\mathrm{cg}}^{-1}\boldsymbol{T}_i^{\mathrm{cg}})^{-1} \tag{53}$$

Similarly, the update equations of the similar states at the nodes j and cg are given as follows

$$\bar{\boldsymbol{x}}_j = \bar{\boldsymbol{P}}_j^{-1}(\boldsymbol{P}_j^{-1}\hat{\boldsymbol{x}}_j + \boldsymbol{T}_i^j\boldsymbol{P}_i^{-1}\hat{\boldsymbol{x}}_i + \boldsymbol{T}_{\mathrm{cg}}^j\boldsymbol{P}_{\mathrm{cg}}^{-1}\hat{\boldsymbol{x}}_{\mathrm{cg}}) \tag{54}$$

$$\bar{\boldsymbol{P}}_j^{-1} = (\boldsymbol{P}_j^{-1} + \boldsymbol{T}_i^j\boldsymbol{P}_i^{-1}\boldsymbol{T}_j^i + \boldsymbol{T}_{\mathrm{cg}}^j\boldsymbol{P}_{\mathrm{cg}}^{-1}\boldsymbol{T}_j^{\mathrm{cg}})^{-1} \tag{55}$$

$$\bar{\boldsymbol{x}}_{\mathrm{cg}} = \bar{\boldsymbol{P}}_{\mathrm{cg}}^{-1}(\boldsymbol{P}_{\mathrm{cg}}^{-1}\hat{\boldsymbol{x}}_{\mathrm{cg}} + \boldsymbol{T}_j^{\mathrm{cg}}\boldsymbol{P}_j^{-1}\hat{\boldsymbol{x}}_j + \boldsymbol{T}_i^{\mathrm{cg}}\boldsymbol{P}_i^{-1}\hat{\boldsymbol{x}}_i) \tag{56}$$

$$\bar{\boldsymbol{P}}_{\mathrm{cg}}^{-1} = (\boldsymbol{P}_{\mathrm{cg}}^{-1} + \boldsymbol{T}_j^{\mathrm{cg}}\boldsymbol{P}_j^{-1}\boldsymbol{T}_{\mathrm{cg}}^j + \boldsymbol{T}_i^{\mathrm{cg}}\boldsymbol{P}_i^{-1}\boldsymbol{T}_{\mathrm{cg}}^i)^{-1} \tag{57}$$

Equations (47)–(50) and (52)–(57) constitute the distributed state fusion filter algorithms for the inertial network system. The state fusion exploits the redundancies of the similar system states and consequently, this fusion method can greatly improve the fault tolerance of an inertial network system.

From (47) and (48), the outputs of each state fusion filter are fed back to the corresponding local Kalman filter. This feedback operation allows the local Kalman filter to accurately estimate and calibrate its sensor errors. This procedure is known as the dynamic transfer alignment of the inertial network system. Both the inertial data fusion and state fusion procedures improve the estimation accuracy of the inertial states and similar states. The local Kalman filters allow all slave node IMUs to be aligned in-flight to the master inertial navigation system.

Compared with the federated filter and other distributed filter architectures [18, 19], the distributed data fusion algorithms presented in this paper afford the following advantages

1. The filter architecture is relatively simple.
2. Each local Kalman filter estimates its own inertial sensor errors rather than pseudo-sensor errors. Therefore the

Table 1: Simulation parameters of inertial sensors

Sensor Parameters	1	2	3	4	5
gyro drift time const, s	110	115	120	100	100
gyro drift err, deg/h	0.1|0.34	0.1|0.3	0.1|0.38	0.1|0.3	0.1|0.45
gyro bias err, deg/h	0.05|10.1	0.05|10.2	0.05|10.1	0.05|10.1	0.05|10.2
gyro SF err time const, s	100|250	100|260	120|270	100|260	104|254
gyro SF error, ppm	50|100	50|100	50|100	50|100	50|100
gyro Az misalign err, arcsec	23	23	23	21	23
gyro El misalign err, arcsec	24	20	21	22	21
gyro noise, deg/sqrt(h)	0.01|0.1	0.01|0.1	0.02|0.08	0.02|0.1	0.01|0.1
accel drift time const, s	60	65	65	60.5	66
accel drift err, μg	50	50	54	50	55
accel bias err, μg	100|310	110|300	100|300	110|300	100|312
accel SF err time const, s	250	260	250	260	245
accel SF err, ppm	100|300	100|300	100|300	100|310	100|300
accel Az misalign err, arcsec	23	23	21	21	23
accel El misalign err, arcsec	22	21	20	22	21
accel noise, μg/sqrt(hz)	30|50	30|50	30|55	30|52	30|50

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

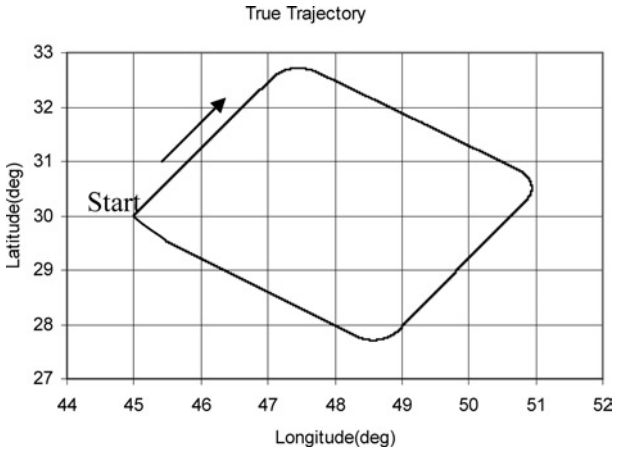

Fig. 4 *True position trajectory*

estimates of the local filters can be used to correct the local IMU errors and the similar states.

3. By assimilating the similar local states, each local state fusion filter provides an inherent fault tolerance capability.

4. The inner feedback from the state fusion filter to the local Kalman filter at each node allows all IMUs in an inertial network system to be automatically aligned to the cg node IMU frame without additional alignment algorithms or procedures. As a result, many traditional inertial alignment procedures, such as transfer alignment, are eliminated.

5 Simulation results

A multi-sensor simulation environment, consisting of a GNSS simulator, an SRIMU simulator and a true trajectory generator, was developed in Matlab to test and evaluate the fusion algorithms described in this paper. The GNSS simulator provides raw GNSS measurements, including pseudoranges and pseudo-range rates at the rate of 1 Hz. The SRIMU simulator can simulate several SRIMU configurations, including cube, cone and dodecahedron configurations. The SRIMU simulator, working with a true trajectory generator, generates realistic SRIMU measurements at an update rate of 50 Hz. The true trajectory is used as a reference to examine the accuracy of the estimated aircraft motion states. The simulated gyro sensors have gyro drift rates of $10^\circ/h$ (a typical tactical-grade IMU) at the slave nodes and $0.05^\circ/h$ (a typical navigation-grade IMU) at the cg node. All simulated accelerometers at the slave nodes have a typical bias of 300 μg, whereas the accelerometers at the cg node have a bias of 100 μg. The SRIMU simulator also models other error sources, including time-dependent sensor drifts (random drifts), zero offset (biases), misalignments, scale factor errors, as well as sensor noise. The parameters used in the simulation are summarised in Table 1. The simulation results show that the change of accelerometer bias has no significant effects on the performance of the master and slave node data fusion filters.

Because the main purpose of this simulation is to evaluate the estimation accuracy of the distributed data fusion algorithms, only two nodes (one master and one slave) were used in the simulation. Additional nodes do not change the architecture of the distributed data fusion filter but significantly increase the complexity of the computation. The dynamic relationship between the master and the slave nodes was simulated by a sinusoidal function.

In the dynamic models of the two-node inertial network system, the similar system states at both the cg node and the slave nodes contain the nine basic navigation state errors (three linear position errors, three velocity errors and three attitude angle errors), which are represented in the different local frames. The local sensor error state vector at the cg node contains accelerometer-related error

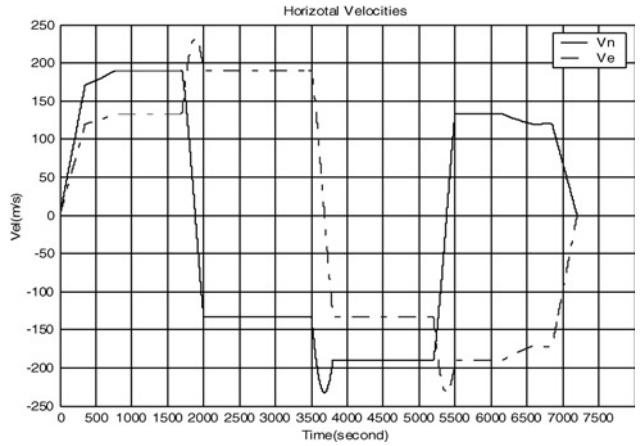

Fig. 5 *Aircraft horizontal manoeuvres*

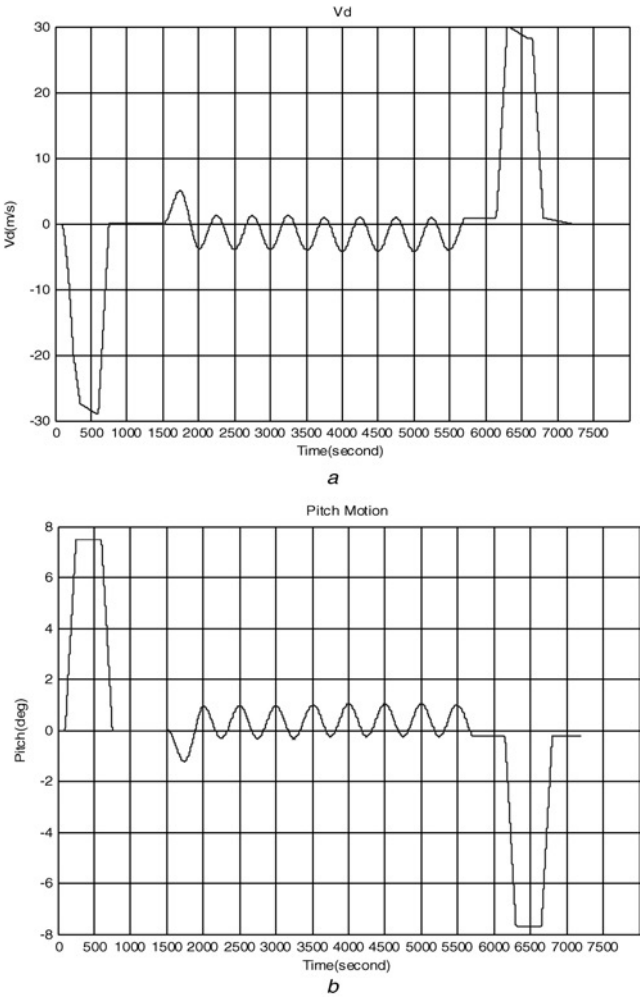

Fig. 6 *Aircraft vertical pitch manoeuvres*

a Vertical velocity manoeuvres
b Pitch manoeuvres

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

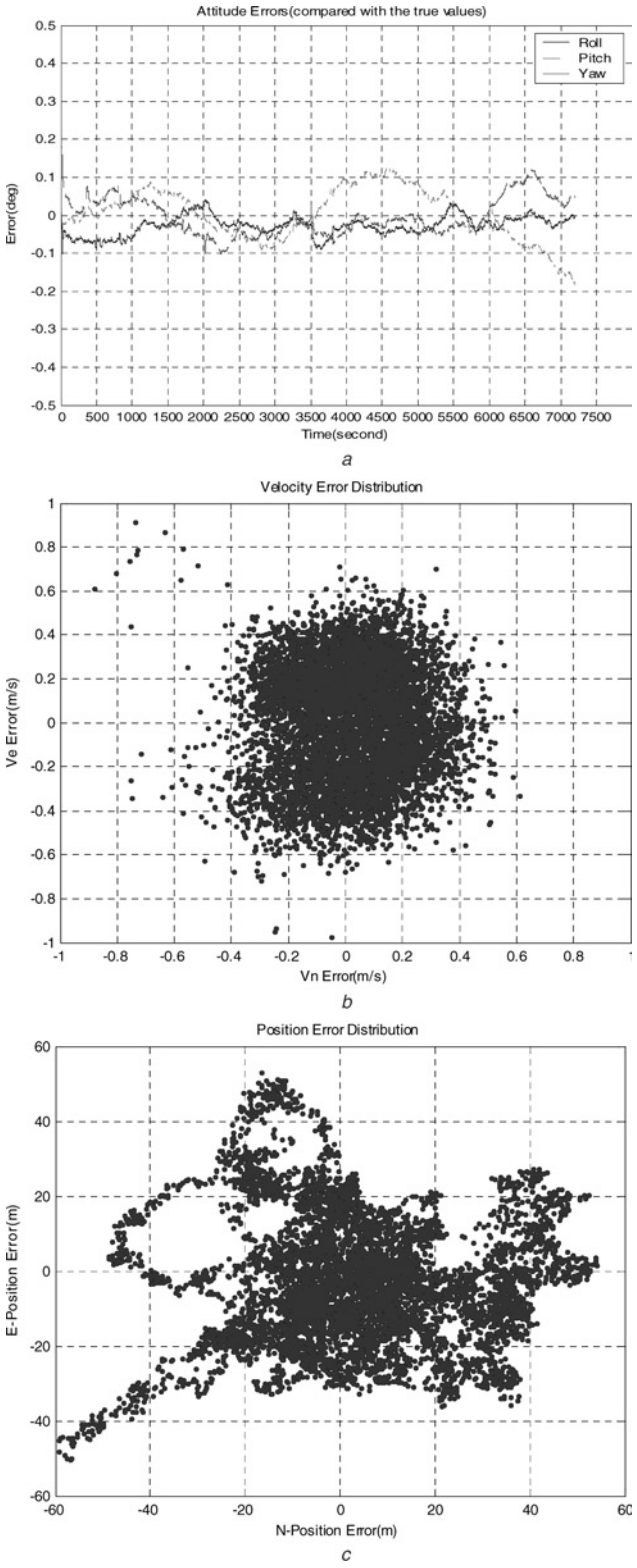

Fig. 7 *Similar state errors at the master node with a gyro drift rate of 0.05°/h*

a Attitude errors at cg node
b Horizontal velocity errors at cg node
c Horizontal position errors at cg node

terms (bias, time-related drift error, misalignments and scale factor error), gyro-related error terms (random constant bias, time-related drift error, misalignments and scale factor error), GNSS receiver clock errors (phase and frequency errors), magnetic heading error and air data system error (air pressure altitude and air speed errors), as specified in [16] and redefined in the following equations.

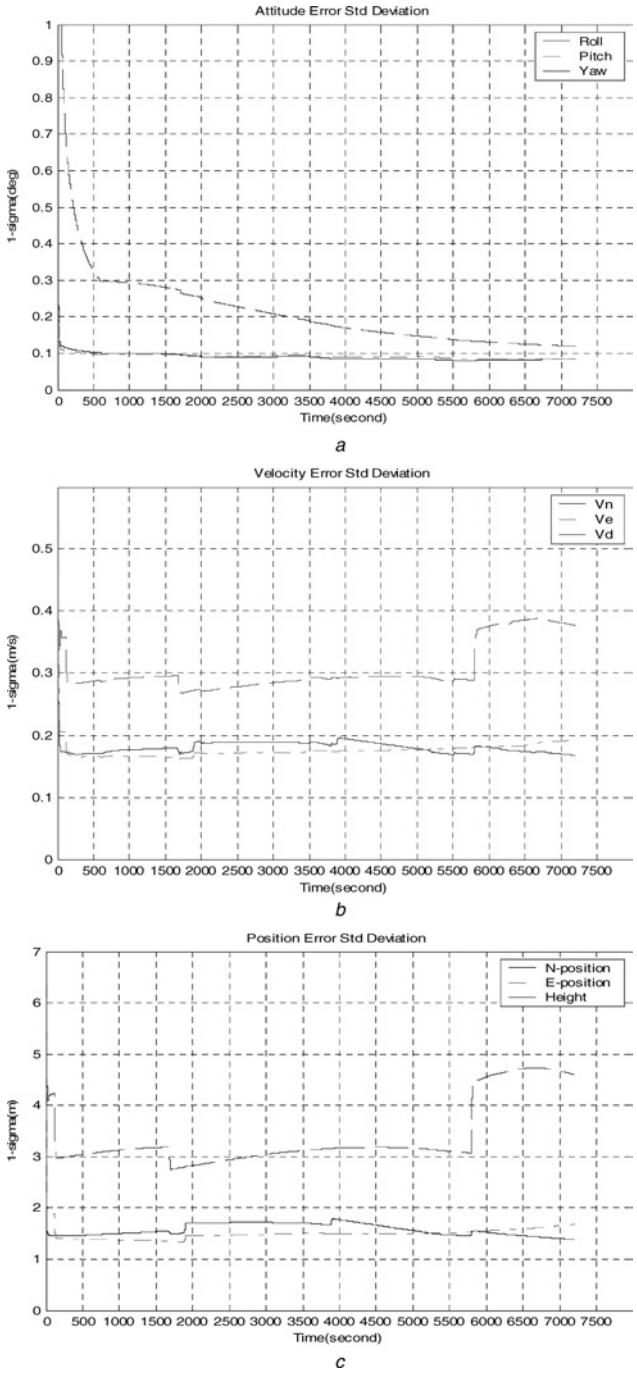

Fig. 8 *Error standard deviations at the master node with a gyro drift rate of 0.05°/h*

a Attitude error standard deviations at cg node
b Velocity error standard deviations at cg node
c Position error standard deviations at cg node

All navigation aiding sensors are related to the cg node. The error state and measurement vectors at the cg node are as follows

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{\text{basic state errors}} \\ \boldsymbol{x}_{\text{sensor errors}} \end{bmatrix}$$

$$\boldsymbol{z} = \begin{bmatrix} \text{GPS pseudorange vector} \\ \text{GPS pseudorange rate vector} \\ \text{magnetic heading output} \\ \text{air pressure altitude and air speed} \\ \text{SRIMU residual vector} \end{bmatrix}$$

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

$$\boldsymbol{x}_{\text{basic state errors}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \delta\varphi_R \\ \delta\lambda_R \\ \delta h \\ \delta v_x^1 \\ \delta v_y^1 \\ \delta v_z^1 \\ \partial\phi_x \\ \partial\phi_y \\ \partial\phi_z \end{bmatrix}$$

$$\boldsymbol{x}_{\text{sensor errors}} = \begin{bmatrix} \text{Rx Clk error states} \\ \text{SRIMU accel error states} \\ \text{SRIMU gyro error states} \\ \text{Magnetic heading error state} \\ \text{Barometer bias state} \\ \text{True air speed bias state} \end{bmatrix}$$

$$= \begin{bmatrix} x_{\text{clk-pha}} \\ x_{\text{clk-rate}} \\ \boldsymbol{x}_{\text{accel-1}} \\ \vdots \\ \boldsymbol{x}_{\text{accel-}n} \\ \boldsymbol{x}_{\text{gyro-1}} \\ \vdots \\ \boldsymbol{x}_{\text{gyro-}n} \\ \boldsymbol{x}_{\text{Mag}} \\ \boldsymbol{x}_{\text{Bar}} \\ \boldsymbol{x}_{\text{ADS}} \end{bmatrix}$$

The local sensor error state vector at the slave node contains only local inertial sensor errors; the measurement vector at the slave node is defined as follows

$$\boldsymbol{z} = \begin{bmatrix} \text{position residual vector} \\ \text{velocity residaul vector} \\ \text{attitude residal vector} \\ \text{SRIMU residual vector} \end{bmatrix}$$

The simulated aircraft can perform arbitrary manoeuvres up to a maximum acceleration of 3.0 g. A typical flight trajectory used in this study is shown in Fig. 4 in which the aircraft flies from the start point to the north-east, then completes three sets of 90° right turns, returning to the start point. Figs. 5 and 6 show the trajectories of the horizontal and vertical manoeuvres, respectively.

The simulation study focused on the assessment of the performance of the distributed data fusion algorithms in terms of estimation accuracy and convergence. Time delay problems in an inertial network system were not considered although they may cause problems in inertial network alignment procedures. In the following case study, both the master and slave nodes use SRIMUs. The simulated IMU at the master node is a 5-sensor cone SRIMU with a gyro drift rate of 0.05°/h. The simulated IMU at the slave node is also a 5-sensor cone SRIMU but with a gyro drift rate of 10°/h.

The simulation results of the similar system states derived by the master node fusion filter are shown in Figs. 7 and 8 where the gyros have a drift rate of 0.05°/h and GPS provides continuous pseudorange and pseudo-range rate observables at the rate of 1 Hz. Fig. 7 shows the absolute errors of the similar system states in comparison with the true trajectory parameters and Fig. 8 shows the error standard

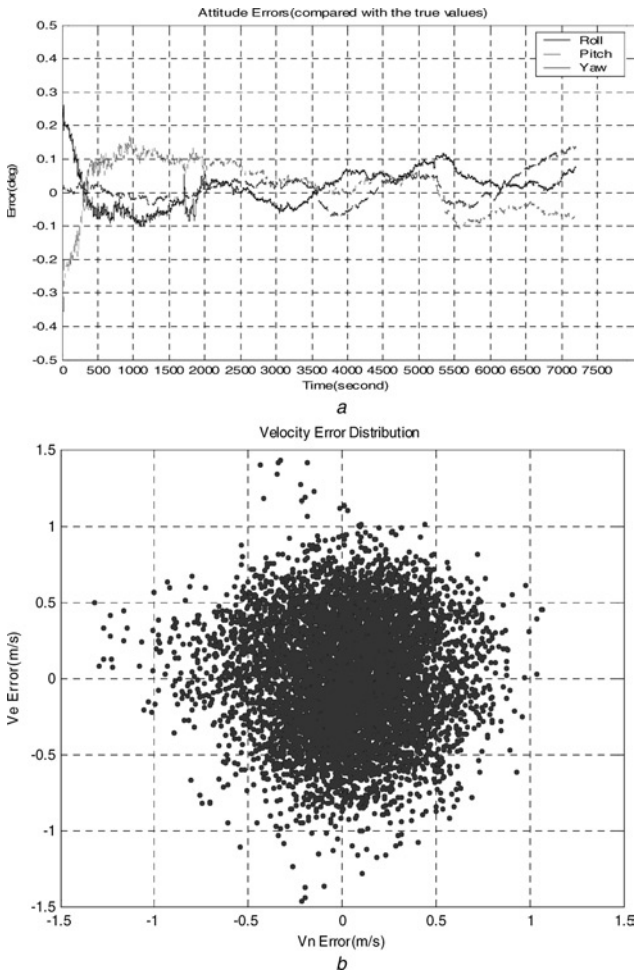

Fig. 9 *Similar state errors at the slave node with a gyro drift rate of 10°/h*

a Attitude errors at slave node
b Horizontal velocity errors at slave node

deviations of the similar system states estimated by the master node fusion filter. After the initial alignment (5 min), the absolute attitude errors (the difference between the true and estimated values) mostly remained within 0.1° during a simulated 2 h flight in Fig. 7*a*. The lateral velocity errors were <0.5 m/s with a probability of 98% in Fig. 7*b* and the accuracy of the lateral velocity estimates is better than 0.2 m/s (σ) in Fig. 8*b*. As expected, the vertical velocity has a larger error compared with the lateral velocity error owing to the vertical dilution of precision of GPS. The cg node data fusion filter quickly converges to the steady state and achieves steady-state attitude estimates <0.1° (1σ) in Fig. 8*a*. The accuracy of the yaw estimate is slightly less than other attitude angles because of a larger initial yaw angle. The positioning accuracy of the master node fusion filter depends on the GPS positioning accuracy. The 'jump' changes of the standard deviation of the position and velocity error states in Figs. 8*b* and *c* were caused by the changes of the GPS satellite geometries.

For the slave node with a gyro drift rate of 10°/h, Fig. 9 shows the absolute errors of the local similar states and Fig. 10 shows the corresponding standard deviations of the local similar states. After the initial alignment (10 min), the accuracy of the attitude estimates at the slave node shows no significant degradation although the convergent rate of the slave node data fusion filter is slower than the master node fusion filter in Fig. 9*a*. The accuracy of the attitude estimate derived by the slave

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

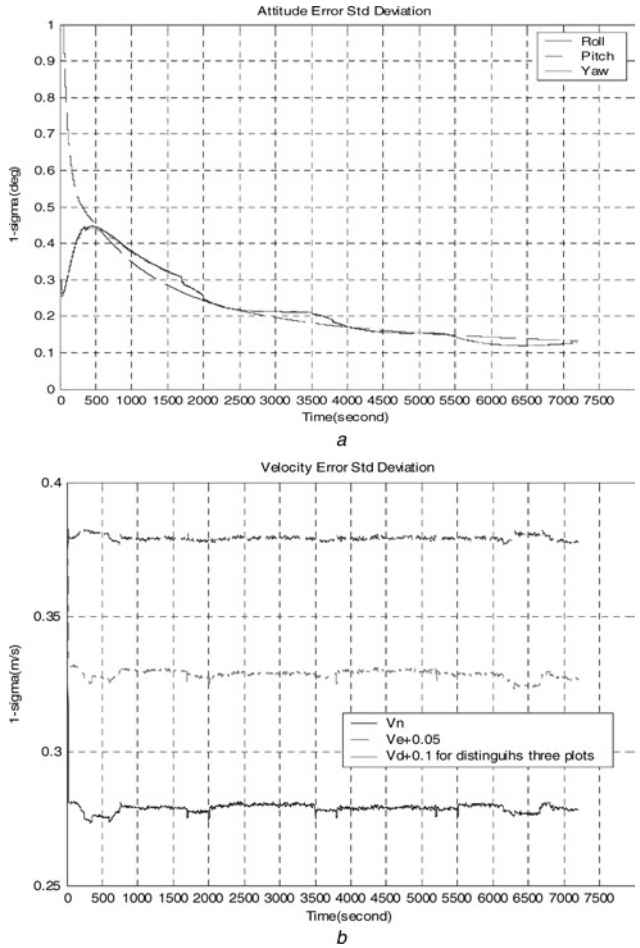

Fig. 10 *Error standard deviations at the slave node with a gyro drift rate of 10°/h*

a Attitude errors standard deviations at slave node
b Velocity errors standard deviations at slave node

fusion filter is better than 0.2° (σ) in Fig. 10*a* and the accuracy of the lateral velocity derived by the slave fusion filter is better than 0.28 m/s (σ) in Fig. 10*b*. From Figs. 9*a* and 10*b*, the estimates of the local similar states change slightly during highly dynamic manoeuvres.

The measured rotation rate of the local pitch axis at the slave node and the corresponding estimate given by the slave fusion filter are shown in Fig. 11. After the initial alignment of the slave IMU (10 min), the accuracy of the estimated pitch rate at the slave node is significantly better than the direct measurements at the slave IMU. These simulations show that the distributed inertial network algorithms presented in this paper can be used to achieve accurate navigation states where the local inertial state vector is derived for low-cost IMUs. It can be seen that the local Kalman filter and the state fusion filter combine to reduce significantly the high-frequency pitch rate noise.

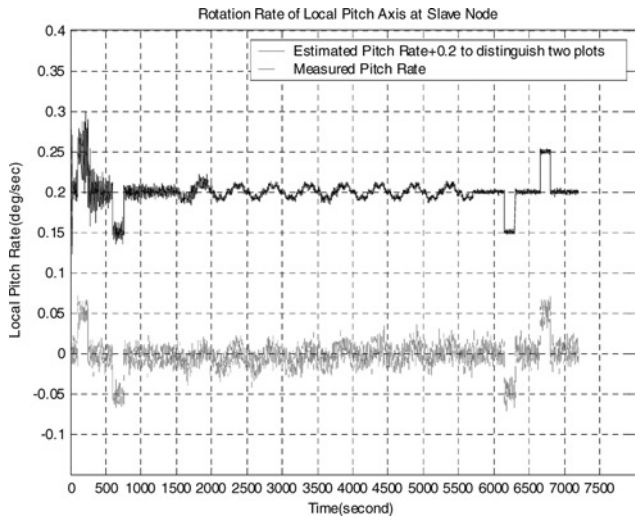

Fig. 11 *Rotation rates of local pitch axis at slave node with gyro drift rate of 10°/h*

6 Conclusions

The paper reviews recent developments in aircraft inertial network systems, with particular emphasis on skew redundant IMUs. The overall system architecture and navigation reference frames are developed for distributed inertial network nodes. Two methods are presented to compute the rotation matrices for the dynamic inertial sensing model for both rigid body aircraft and flexible airframe configurations.

Distributed data fusion algorithms have been developed for inertial network systems, including the development of distributed state fusion filters and inertial measurement fusion algorithms. These algorithms are applicable to multiple IMU network systems in aircraft with a flexible structure. Extensive simulation studies were undertaken combining an SRIMU simulator with a GNSS simulator. Specific manoeuvres were flown so that the sensor derived measurements and aircraft motion states could be compared directly with the actual aircraft trajectory.

Simulation studies have demonstrated the feasibility of distributed data fusion algorithms for local inertial state estimate and dynamic inertial network alignment. A wide range of low-cost inertial sensors were evaluated, covering the performance of the majority of currently available inertial sensors. The simulation results presented in the paper show that low-quality IMUs (gyro drift rate up to 30°/h) can be precisely aligned to a high-quality IMU (gyro drift rate of 0.05°/h) with an alignment accuracy better than 0.2°/h by using the distributed data fusion filters and that the local fusion filter can accurately estimate the local inertial states. The algorithms presented in this paper provide redundant inertial information in a form that is applicable to current failure detection and isolation methods used to detect sensor failures.

7 References

1 Kelley, R.T., Carlson, N.A., and Berning, S.: 'Integrated inertial network'. Proc. IEEE PLANS, 1994, pp. 439–446
2 Berning, S., Howe, P., and Jenkins, T.: 'Theater-wide reference information management'. Proc. IEEE NAECON, 1996, pp. 122–128
3 Kaiser, J., Beck, G., and Berning, S.: 'Vital advanced inertial network'. Proc. IEEE PLANS, 1998, pp. 61–68
4 Harris, R.L.: 'Modular avionics: its impacts on communication, navigation, and identification (CNI)'. Proc. IEEE NAECON, 1988, pp. 1164–1169
5 Swanson, D.L.: 'Evolving avionics systems from federated to distributed architectures'. Proc. IEEE/AIAA/NASA 17th Digital Avionics Systems Conf., 1998, pp. D26/1–D26/8
6 Schmidt, G.T.: 'GPS/INS technology trends for military systems'. Draper Technology Digest, 1998, pp. 2–13 available at: http://www.draper.com/pubns/digest98/digest98.htm
7 Barbour, N. Schmidt, G.T.: 'Inertial Sensor Technology Trends', *IEEE Sensors J.*, **1**, (4)Dec 2001, pp. 332–339
8 Gilmore, J.P., and Mckern, R.A.: 'A redundant strapdown inertial reference unit (SIRU)', *J. Spacecraft Rockets*, 1972, **9**, (1), pp. 39–47
9 Allerton, D.J., and Jia, H.: 'An error compensation method for skewed redundant inertial configurations'. Proc. ION 58th Annual Meeting/GIGTF 21st Guidance Test Symp., Albuquerque, USA, 2002, pp. 142–147

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

10 Daly, K.C., Gai, E., and Harrison, J.V.: 'Generalized likelihood ratio test for FDI in redundant sensor configurations', *J. Guid. Control*, 1979, **2**, (1), pp. 9–17
11 Brown, A., and Sturza, M.A.: 'The effect of geometry on integrity monitoring performance'. Proc. ION 46th Annual Meeting, 26–28 June 1990
12 Carison, N., Kelley, R., and Berning, S.: 'Differential inertial filter for dynamic sensor alignment'. Proc. ION National Technical Meeting, 1994, pp. 341–351
13 Maybeck, P.S.: 'Stochastic models, estimation, and control' (Academic Press, 1979), vol. 1
14 Hanlon, P.D., and Maybeck, P.S.: 'Multiple-model adaptive estimation using a residual correlation kalman filter bank', *IEEE Trans. Aerosp. Electron. Syst.*, 2000, **36**, (2), pp. 393–406
15 Roger, R.M.: 'Applied mathematics in integrated navigation systems' (AIAA, Inc., 2000)
16 Allerton, D.J., and Jia, H.: 'Redundant multi-mode filter for a navigation system', *IEEE Trans. Aerosp. Electron. Syst.*, 2007, **43**, (1), pp. 371–391
17 Jia, H.: 'Data fusion methodologies for multisensor aircraft navigation systems', PhD thesis, Cranfield University, Bedfordshire, UK, 2004
18 Carlson, N.A.: 'Federated filter for distributed navigation and tracking applications'. Proc. ION 58th Annual Meeting/GIGTF 21st Guidance Test Symp., Albuquerque, USA, 2002, pp. 142–147
19 Allerton, D.J., and Jia, H.: 'A Review of multisensor fusion methodologies for aircraft navigation systems', *J. Navig.*, 2005, **58**, (3), pp. 405–417

Authorized licensed use limited to: IEEE Xplore. Downloaded on January 19, 2009 at 11:31 from IEEE Xplore. Restrictions apply.

https://dspace.org/

Distributed data fusion algorithms for inertial network systems

Allerton, David J.; Jia, Huamin

2008-01-01T00:00:00Z
Institution of Engineering and Technology (IET)
http://dx.doi.org/10.1049/iet-rsn:20060159
Allerton DJ, Jia H. (2008) Distributed data fusion algorithms for inertial network systems. IET Radar, Sonar & Navigation, Volume 2, Issue 1, February 2008, pp. 51-62

Downloaded from a DSpace repository. https://dspace.org/