

Article

Data Fusion Architectures for Orthogonal Redundant Inertial Measurement Units

Eric Gagnon ^{1,*}, Alexandre Vachon ² and Yanick Beaudoin ³

¹ Defence Research and Development Canada, Quebec, QC G3J 1X5, Canada

² Numérica Technologies Inc., Quebec, QC G2E 4P8, Canada;
Alexandre.Vachon.NUMERICA@drdc-rddc.gc.ca

³ Département de génie électrique et génie informatique, Université Laval, Quebec, QC G1V 0A6, Canada;
yanick.beaudoin.1@ulaval.ca

* Correspondence: Eric.Gagnon@drdc-rddc.gc.ca; Tel.: +1-418-844-4000 (ext. 4908)

Received: 18 May 2018; Accepted: 8 June 2018; Published: 12 June 2018



Abstract: This work looks at the exploitation of large numbers of orthogonal redundant inertial measurement units. Specifically, the paper analyses centralized and distributed architectures in the context of data fusion algorithms for those sensors. For both architectures, data fusion algorithms based on Kalman filter are developed. Some of those algorithms consider sensors location, whereas the others do not, but all estimate the sensors bias. A fault detection algorithm, based on residual analysis, is also proposed. Monte-Carlo simulations show better performance for the centralized architecture with an algorithm considering sensors location. Due to a better estimation of the sensors bias, the latter provides the most precise and accurate estimates and the best fault detection. However, it requires a much longer computational time. An analysis of the sensors bias correlation is also done. Based on the simulations, the biases correlation has a small effect on the attitude rate estimation, but a very significant one on the acceleration estimation.

Keywords: orthogonal redundant inertial measurement units; data fusion architectures; sensors bias

1. Introduction

In the beginning of the 90s, a novel sensor design technique changed the manufacturing and operation of inertial navigation systems (INS). Micro-electro-mechanical systems (MEMS) integrate the classical mechanical design in the integrated circuit fabrication technology. This new class of small inertial measurement units (IMUs) can then be fabricated in large batches, drastically reducing the production costs. However, their smaller size is problematic; a smaller sensor is more sensitive to temperature and has a smaller scale factor, resulting in a lower signal-to-noise ratio than classical IMUs. On the other hand, the MEMS' small size and low cost make them suitable to build arrays of redundant sensors for fused measurements objective. The combination of inertial sensors in arrays has therefore become an important field of research lately [1].

The most common purpose for redundant inertial measurement units (RIMUs) in navigation systems is to facilitate the detection and isolation of faulty sensors [2]. The precision and accuracy improvement provided by using multiple measurements is often a secondary objective. In this optic, the studied sensors are, more than often, positioned in a non-orthogonal skewed configuration, known as SRIMU, or on regular polyhedrons. In a SRIMU, the sensors are distributed on a cone, in a way that there is a constant angle between two consecutive sensors [3]. This configuration provides the maximum redundancy and, hence, encapsulates the maximum amount of information for a specific number of IMUs [4]. The regular polyhedrons are, on their parts, the optimal configuration for navigation and fault detection and isolation (FDI) [5].

However, these configurations are interesting only for a relatively low number of redundancies and for costly sensors [6]. When dealing with a large number of IMUs, these optimal configurations are very complex and the interest of lowering the cost of the redundancy by using the minimal number of IMUs is overshadowed by the cost due to the complexity of the assembly. In this case, an orthogonal configuration, where triads of sensors are mounted parallel one to the others, can be interesting. This configuration is much easier to assemble, but, in comparison to the SRIMU and polyhedrons configurations, has poor FDI performance [6]. In order to provide the same level of robustness (same number of detected faults), a higher number of IMUs is required in the orthogonal configuration. However, for RIMU based on MEMS, which are relatively cheap and small, this is not an issue. As a result, the orthogonal configuration should be seriously considered for those sensors [6].

The handling of these redundant measurements can be categorised in two types: centralized or distributed. In the centralized architecture, all sensor measurements are directly processed by a single central data fusion filter in order to obtain the state estimates. This architecture is the optimal [7] and most common design in current INS [8], even though it is not necessarily the most efficient one. With multiple sensors of different types, the centralized filter computation might be time-consuming [8], and it is not the ideal method for a fault tolerant multi-sensors scheme [9]. However, when only IMUs are used, the centralized architecture remains time-consuming, but does not exhibit FDI sub-performance [2].

On its part, the distributed architecture is a multi-stage architecture where parallel local filters independently process the measurements of their corresponding sensors and their outputs are used by a master data fusion filter which produces the final state estimates. Also, the master filter can send back information to the local filters in order to reset some components of their internal state. As the centralized architecture, this architecture can offer optimal performance, but only when all filters model the full state and when they are all ran at the same rate [7]. The main advantage of this architecture is its better FDI performance [9]. This architecture has been applied to multiple types of multi-sensors navigation systems [9,10] and to multiple IMUs systems [2,4].

For vehicle navigation, the fusion of IMUs is mainly studied as part of navigation algorithm aided by Global Positioning System (GPS). However, some works [11–13] discard the GPS and rely on alternative (radar or vision) measurement devices, while others rely purely on inertial sensors, accelerometers only [14,15] or accelerometers and gyroscopes [16].

Considering the fictitious forces due to sensors location is not common in RIMU data fusion algorithms, but has been done [2,16]. Similarly, estimating the sensors bias within the algorithms has been considered [17–19]. It is also the case for the analysis of correlation between the sensor random walk noises [20,21], but not a correlation between the biases of the sensors.

This paper intends to simultaneously address those three aspects (sensors locations, biases estimation and biases correlation) within navigation based on IMUs only. Above all, the paper studies their effects on each type of data fusion architecture. The main objective is the identification of the ideal architecture for data fusion of a large number of orthogonal RIMU. For the two studied architectures, data fusion algorithms based on Kalman filter are used (Section 2). Those algorithms, notably, estimate the sensors bias. For both architectures, an algorithm which considers sensors location, and one which does not, are proposed. A baseline algorithm, averaging the measurements, is used as a comparison point. A fault detection and isolation algorithm, based on residual statistical analysis, is also presented (Section 3). Those algorithms are then compared using Monte-Carlo simulations (Section 4). The simulated application consists in an artillery shell (spin stabilized projectile) equipped with strapdown orthogonal RIMU having, as a first case, uncorrelated biases and, as a second case, correlated ones. A simulations-based analysis is mandatory for this application; it would have been impossible to test the same amount of sensors locations, noises, biases and faults in the field.

2. Data Fusion Algorithms

This section proposes data fusion algorithms based on Kalman filter for MEMS RIMU. MEMS sensors are characterized by large random walk and bias instability [22]. The random walk is, by nature, a white noise which can be efficiently processed by a Kalman filter, but the bias instability is a second order random walk [22] and requires a specific treatment in order to be estimated by a Kalman filter.

Obviously, it is impossible, even with MEMS, to have perfectly collocated sensors. The distance between the IMUs can be very small and much smaller than the sensitivity of the sensors, but this distance still influence the quality of the estimations. The Grubin transformation [23], which expresses the fictitious forces applied on a point not located at the centre of mass of a body, is used to include the effect of the angular motion on the accelerometer measurements by taking into account the sensors location in the models. A triad of accelerometers, located at the sensor position (s_{sb}), then measures three accelerations: the nominal linear acceleration (a), the angular acceleration ($\dot{\omega} \times s_{sb}$) and the centrifugal acceleration ($\omega \times (\omega \times s_{sb})$). On its part, the gyroscopes triad is not affected by its location and still measures the angular velocity (ω) only.

For the centralized and distributed architectures, the Kalman filter prediction model required to consider the sensors location will be presented. Then, by eliminating the fictitious forces, a model for an algorithm not considering sensors location will be derived. Similarly, for the baseline architecture, the algorithm discarding sensors location is derived from the algorithm considering sensors location.

2.1. Centralized Architecture

In a typical GPS/INS navigation solution, the coupling between the acceleration and attitude rate measurements appears, through the rotation matrix, in the velocity and position vectors [24]. However, without GPS receiver, the velocity, position and attitude are unobservable, and can, without performance degradation, be eliminated from the data fusion algorithm in order to decrease its complexity. For the centralized architecture, the filter state is then composed of the acceleration estimates (a_e), the attitude rate estimates (ω_e) and the biases of all accelerometers (b_a) and of all gyroscopes (b_ω):

$$x = \left[a_e \quad b_{a1} \quad \cdots \quad b_{an} \quad \omega_e \quad b_{\omega 1} \quad \cdots \quad b_{\omega n} \right]^T \quad (1)$$

On its part, the state propagation equation, which is linear, is:

$$x_{k+1} = \begin{bmatrix} a_e \\ b_{a1} \\ \vdots \\ b_{an} \\ \omega_e \\ b_{\omega 1} \\ \vdots \\ b_{\omega n} \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \cdots & \mathbf{0}_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{I}_3 \end{bmatrix} x_k + \begin{bmatrix} v_a \\ v_{ba1} \\ \vdots \\ v_{ban} \\ v_\omega \\ v_{b\omega 1} \\ \vdots \\ v_{b\omega n} \end{bmatrix}_k \quad (2)$$

and the nonlinear output equation is:

$$y_k = \begin{bmatrix} a_{m1} \\ a_{m2} \\ \vdots \\ a_{mn} \\ \omega_{m1} \\ \omega_{m2} \\ \vdots \\ \omega_{mn} \end{bmatrix}_{k+1} = h(x_k) = \begin{bmatrix} a_e + \dot{\omega}_e \times s_{sb1} + \omega_e \times (\omega_e \times s_{sb1}) + b_{a1} \\ a_e + \dot{\omega}_e \times s_{sb2} + \omega_e \times (\omega_e \times s_{sb2}) + b_{a2} \\ \vdots \\ a_e + \dot{\omega}_e \times s_{sbn} + \omega_e \times (\omega_e \times s_{sbn}) + b_{an} \\ \omega_e + b_{\omega 1} \\ \omega_e + b_{\omega 2} \\ \vdots \\ \omega_e + b_{\omega n} \end{bmatrix} + \begin{bmatrix} w_{a1} \\ w_{a2} \\ \vdots \\ w_{an} \\ w_{\omega 1} \\ w_{\omega 2} \\ \vdots \\ w_{\omega n} \end{bmatrix}_k \quad (3)$$

The Kalman filter state propagation matrix is the identity matrix of Equation (2), while its output matrix is the Jacobian of Equation (3):

$$C_k = \frac{\partial h}{\partial x} \Big|_k = \begin{bmatrix} I_3 & I_3 & 0_3 & \cdots & 0_3 & a_{\omega 1} & 0_3 & 0_3 & \cdots & 0_3 \\ I_3 & 0_3 & I_3 & \cdots & 0_3 & a_{\omega 2} & 0_3 & 0_3 & \cdots & 0_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_3 & 0_3 & 0_3 & \cdots & I_3 & a_{\omega n} & I_3 & 0_3 & \cdots & 0_3 \\ 0_3 & 0_3 & 0_3 & \cdots & 0_3 & I_3 & I_3 & 0_3 & \cdots & 0_3 \\ 0_3 & 0_3 & 0_3 & \cdots & 0_3 & I_3 & 0_3 & I_3 & \cdots & 0_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \ddots & 0_3 \\ 0_3 & 0_3 & 0_3 & \cdots & 0_3 & I_3 & 0_3 & 0_3 & \cdots & I_3 \end{bmatrix} \quad (4)$$

where the a_{ω_i} matrix is the Jacobian of the i th accelerometer measurements with respect to the attitude rate estimates:

$$a_{\omega_i} = \begin{bmatrix} q_e s_{y_i} + r_e s_{z_i} & p_e s_{y_i} - 2q_e s_{x_i} & p_e s_{z_i} - 2r_e s_{x_i} \\ q_e s_{x_i} - 2p_e s_{y_i} & p_e s_{x_i} + r_e s_{z_i} & q_e s_{z_i} - 2r_e s_{y_i} \\ r_e s_{x_i} - 2p_e s_{z_i} & r_e s_{y_i} - 2q_e s_{z_i} & p_e s_{x_i} + q_e s_{y_i} \end{bmatrix} \quad (5)$$

where p_e , q_e and r_e , the components of ω_e , are the estimates of, respectively, the roll, pitch and yaw rates, and s_{x_i} , s_{y_i} and s_{z_i} , the components of s_{sb_i} , are the locations of the i th accelerometer along, respectively, the body frame x , y and z axes, with respect to the body centre of mass.

Assuming that all sensors have the same specifications, the state noise covariance matrix (Q) and the measurement noise matrix (R) of the Kalman filter are:

$$Q = \begin{bmatrix} \sigma_{ae}^2 & 0_3 & \cdots & 0_3 & 0_3 & 0_3 & \cdots & 0_3 \\ 0_3 & \sigma_{ba1}^2 I_3 & \cdots & 0_3 & 0_3 & 0_3 & \cdots & 0_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_3 & 0_3 & \cdots & \sigma_{ban}^2 I_3 & 0_3 & 0_3 & \cdots & 0_3 \\ 0_3 & 0_3 & \cdots & 0_3 & \sigma_{\omega e}^2 & 0_3 & \cdots & 0_3 \\ 0_3 & 0_3 & \cdots & 0_3 & 0_3 & \sigma_{b\omega 1}^2 I_3 & \cdots & 0_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_3 & 0_3 & \cdots & 0_3 & 0_3 & 0_3 & \cdots & \sigma_{b\omega n}^2 I_3 \end{bmatrix} \quad (6)$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{wa1}^2 \mathbf{I}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \cdots & \sigma_{wan}^2 \mathbf{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \sigma_{w\omega 1}^2 \mathbf{I}_3 & \cdots & \mathbf{0}_3 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \sigma_{w\omega n}^2 \mathbf{I}_3 \end{bmatrix} \quad (7)$$

where σ_{bai}^2 is the variance of the i th accelerometer bias, $\sigma_{b\omega i}^2$ is the variance of the i th gyroscope bias, σ_{wai}^2 is the variance of the i th accelerometer random walk and $\sigma_{w\omega i}^2$ is the variance of the i th gyroscope random walk. The matrices σ_{ae}^2 and $\sigma_{\omega e}^2$ contain the variances of the jerk and angular acceleration, respectively. The variances of the three axes are expressed, individually, along the diagonals of the matrices. The variances of the jerk and of the angular acceleration are estimated based on the expected movements of the sensor, by computing the maximum variation in the expected consecutive measurements. Artillery munitions trajectory is predictable and a good estimation of this variance is achievable. However, for vehicles having unpredictable trajectory (e.g., missiles), the estimation would not be as good.

On their parts, the bias and random walk variances are set to mimic the sensor specifications. As the same sensors are used in all IMUs, all the accelerometers random walk variances ($\sigma_{wa1\dots n}^2$) are equal. This is also the case for the accelerometers bias variances ($\sigma_{ba1\dots n}^2$), for the gyroscopes random walk variances ($\sigma_{w\omega 1\dots n}^2$) and for the gyroscopes bias variances ($\sigma_{b\omega 1\dots n}^2$).

As they cannot be measured, the angular accelerations ($\dot{\omega}$) must be estimated. This can be done by additional components in the Kalman filter state, or by backward finite differences (Equation (8)). The latter is selected in order to decrease the size of the filter state, hence the computational time:

$$\dot{\omega}_{ek} = \begin{bmatrix} \dot{p}_e \\ \dot{q}_e \\ \dot{r}_e \end{bmatrix} = \frac{1}{\delta t} \begin{bmatrix} p_{ek-1} - p_{ek-2} \\ q_{ek-1} - q_{ek-2} \\ r_{ek-1} - r_{ek-2} \end{bmatrix} \quad (8)$$

For algorithm not considering sensors location, the fictitious forces are null. The term $\dot{\omega}_e \times s_{sbi} + \omega_e \times (\omega_e \times s_{sbi})$ in Equation (3) and the $\mathbf{a}_{\omega i}$ matrix in Equation (4) are then null. There is therefore no relation between the accelerometer and gyroscope measurements, and no coupling between the axes. Then, in order to diminish the computational time, all axes and sensor types are processed separately. The algorithm not considering sensors location is therefore split in three data fusion filters for the acceleration estimation and three others for the attitude rate estimation. The model of each filter is a subset of the complete model (Equations (2) and (3)); it corresponds to the rows and lines of the estimated parameter and the related sensors bias. The model is then:

$$\mathbf{x}_{k+1} = \begin{bmatrix} a_e \\ b_{a1} \\ b_{a2} \\ \vdots \\ b_{an} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} v_a \\ v_{ba1} \\ v_{ba2} \\ \vdots \\ v_{ban} \end{bmatrix}_k \quad (9)$$

$$\mathbf{y}_k = \begin{bmatrix} a_{m1} \\ a_{m2} \\ \vdots \\ a_{mn} \end{bmatrix}_k = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} w_{a1} \\ w_{a2} \\ \vdots \\ w_{an} \end{bmatrix}_k \quad (10)$$

and the covariance matrices are:

$$\mathbf{Q} = \begin{bmatrix} \sigma_{ae}^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{ba_1}^2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{ba_2}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{ba_n}^2 \end{bmatrix} \quad (11)$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{wa_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{wa_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{wa_n}^2 \end{bmatrix} \quad (12)$$

The previous equations are for the accelerometers. The processing of the gyroscopes is identical, but with the accelerations being replaced by the attitude rates and their corresponding standard deviations. The models of the six filters are then identical. The only difference resides in the values within the covariance matrices, which are related to the estimated parameter.

In the filters initial state, the acceleration and attitude rate estimates are set based on the expected launching state, and the sensors bias are set at 0. On its part, the initial error covariance matrix is set equal to the state noise matrix.

2.2. Distributed Architecture

For the distributed architecture, the state of each local filter is composed of 12 elements:

$$\mathbf{x} = \begin{bmatrix} \mathbf{a}_e & \boldsymbol{\omega}_e & \mathbf{b}_a & \mathbf{b}_\omega \end{bmatrix}^T \quad (13)$$

The state equation is therefore:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{a}_e \\ \boldsymbol{\omega}_e \\ \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_\omega \\ \mathbf{v}_{ba} \\ \mathbf{v}_{b\omega} \end{bmatrix}_k \quad (14)$$

while the output equation is:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{a}_m \\ \boldsymbol{\omega}_m \end{bmatrix}_{k+1} = \mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \mathbf{a}_e + \dot{\boldsymbol{\omega}}_e \times \mathbf{s}_{sb} + \boldsymbol{\omega}_e \times (\boldsymbol{\omega}_e \times \mathbf{s}_{sb}) + \mathbf{b}_a \\ \boldsymbol{\omega}_e + \mathbf{b}_\omega \end{bmatrix} + \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_\omega \end{bmatrix}_k \quad (15)$$

A Kalman filter is implemented with its output matrix being the Jacobian of Equation (15):

$$\mathbf{C}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_k = \begin{bmatrix} \mathbf{I}_3 & \mathbf{a}_\omega & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad (16)$$

Assuming that all sensors have the same specifications, its covariance matrices are:

$$\mathbf{Q} = \begin{bmatrix} \sigma_{ae}^2 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{\omega_e}^2 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \sigma_{ba}^2 \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \sigma_{b\omega}^2 \mathbf{I}_3 \end{bmatrix} \quad (17)$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{\omega a}^2 \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{\omega \omega}^2 \mathbf{I}_3 \end{bmatrix} \quad (18)$$

As for the centralized architecture, the angular accelerations cannot be measured and are estimated using backward finite differences (Equation (8)).

For the algorithm not considering sensors location, as for the centralized architecture, the fictitious forces are null and, for each sensor, the data fusion process can be separated in six filters handling the sensor types and axes individually. When the sensors location is not considered, the subset model used in each filter is composed of the rows and lines of Equations (14)–(16) corresponding to the estimated parameter and its related sensor bias:

$$\mathbf{x}_{k+1} = \begin{bmatrix} a_e \\ b_a \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} v_a \\ v_{ba} \end{bmatrix}_k \quad (19)$$

$$\mathbf{y}_k = a_{mk} = \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x}_k + w_{ak} \quad (20)$$

and the covariance matrices are:

$$\mathbf{Q} = \begin{bmatrix} \sigma_{a_e}^2 & 0 \\ 0 & \sigma_{b_a}^2 \end{bmatrix} \quad (21)$$

$$\mathbf{R} = \sigma_{\omega a}^2 \quad (22)$$

The previous equations are for the accelerometers. The processing of the gyroscopes is identical, but with the accelerations being replaced by the attitude rates and their corresponding standard deviations.

The state of each local filter is initialized based on the expected launching state. The sensors bias are set at 0, and each local filter error covariance matrix is set identical to the state noise matrix.

For both cases, considering or not the sensors location, following the local filters computation, a master data fusion algorithm computes the mean of all estimations for each parameter. Those means are the final estimates, which are sent to the local filters to reset their estimates before the next time step.

2.3. Baseline Architecture

In order to better assess the performance of two previous architectures, a baseline architecture is proposed. The baseline architecture computes for each axis (x , y and z) and sensor type (gyroscope and accelerometer), the measurements average. However, because of the centrifugal and angular accelerations, pre-computations are required on the accelerometer measurements before computing their means.

First, the angular velocity is obtained by directly computing, on each axis separately, the mean of the gyroscope measurements.

The acceleration and attitude rate estimations are then used to remove the angular and centrifugal accelerations from the accelerometer measurements:

$$\mathbf{a}_i = \mathbf{a}_{mi} - (\dot{\boldsymbol{\omega}}_e \times \mathbf{s}_{sb}) - (\boldsymbol{\omega}_e \times (\boldsymbol{\omega}_e \times \mathbf{s}_{sb})) \quad (23)$$

As previously, the angular accelerations, required in the linear accelerations computation, are estimated by backward finite differences (Equation (8)).

The acceleration estimates are the means of the Equation (23) subtractions:

$$a_{x_e} = \frac{\sum_{i=1}^n (a_{mxi} - (p_e q_e s_{y_i} - q_e^2 s_{x_i} + p_e r_e s_{z_i} - r_e^2 s_{x_i}) - (\dot{q}_e s_{z_i} - \dot{r}_e s_{y_i}))}{n} \quad (24)$$

$$a_{y_e} = \frac{\sum_{i=1}^n (a_{my_i} - (p_e q_e s_{x_i} - p_e^2 s_{y_i} + q_e r_e s_{z_i} - r_e^2 s_{y_i}) - (\dot{r}_e s_{x_i} - \dot{p}_e s_{z_i}))}{n} \quad (25)$$

$$a_{ze} = \frac{\sum_{i=1}^n (a_{mzi} - (p_e r_{eS_{xi}} - p_e^2 s_{zi} + q_e r_{eS_{yi}} - q_e^2 s_{zi}) - (\dot{p}_e s_{yi} - \dot{q}_e s_{xi}))}{n} \quad (26)$$

For the algorithm not considering sensors location, the fictitious forces terms can be removed from the previous equations. The acceleration estimates then become the average, on each axis separately, of the accelerometers measurements only.

3. Fault Detection and Isolation Algorithm

The FDI algorithm, identical for the three architectures, is based on a statistical analysis of the residual. There are systematic ways to deal with faulty sensors [6,25], but as this work realizes relative comparisons of data fusion architectures, a simple FDI method is used.

The residual is the difference between the sensor measurement, and the parameter and sensor bias estimates:

$$r_{ax_1} = a_{x1} - (a_{xe} + b_{ax_1}) \quad (27)$$

For the baseline architecture, where the bias is not estimated, the latter is considered null in the residual computation.

If this residual is larger than four times the standard deviation of the sensor random walk for two consecutive measurements, the sensor is considered faulty. The two consecutive 4σ threshold is selected in order to limit false warnings, while ensuring detection of faulty sensors. The sensor random walk being considered as a normal distribution, there is 0.0063% (1 in 15,787) chance that a healthy sensor gives a value larger than the threshold. With a sample time of 0.001 s, this is a false warning every 16 s. For two consecutive measurements, this percentage drops to $4.01 \times 10^{-9}\%$ (1 in 249,229,369), which is approximately a false warning every 29 days. This percentage is equivalent to a single value larger than a threshold of 5.9σ , approximately. However, because of the exponential tail of the normal distribution, a faulty sensor, which is a sensor with a random walk of standard deviation larger than the nominal sensor, has more chance of producing two consecutive measurements larger than 4σ than a single measurement larger than 5.9σ .

4. Performance Analysis

The architectures and algorithms are tested, in simulation, on a spin-stabilized projectile. A typical low quadrant elevation, low muzzle velocity and Northward launch trajectory is obtained from a projectile simulator developed in a non-spinning body frame [23]. The sensors measurements are simulated by adding random noises and bias instabilities to the nominal accelerations and attitude rates. Except when otherwise mentioned, the noises and biases are completely uncorrelated.

Two series of tests are done. First, the quality of the estimation of each architecture and algorithm, without faulty sensors, is tested (Section 4.1). Then, the performance of the fault detection algorithm is analysed (Section 4.2). For both series, the sensors are distributed around the projectile centre of mass. Hence, each accelerometer measures a different acceleration based on the sensed fictitious forces at its location.

4.1. Estimation Precision and Accuracy

As non-collocated sensors are used, the distance between the IMUs, their relative locations and their numbers influence the quality of the estimation of each algorithm. These three parameters are separately analysed.

4.1.1. Relative Locations of the IMUs

The effects of the relative locations of the IMUs are studied by comparing a symmetric 27 IMUs configuration to a configuration where the 27 IMUs are randomly located. In the random configuration, the sensors are all located within the cube of the symmetric configuration (Figure 1) and the randomly

chosen positions are kept for all simulations. The random configuration is physically impracticable, but it is interesting to evaluate the sensors dissymmetry effect on the algorithms.

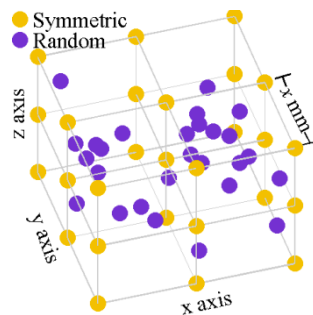


Figure 1. Symmetric and random 27 IMUs configurations.

Monte-Carlo simulations, composed of 350 runs each, are executed for 10 and 100 mm separations between the symmetric IMUs (20 and 200 mm edges cubes), and for their corresponding random locations. As the gyroscope measurements are not affected by their locations, the estimation of the attitude rate is not affected by the random configuration. However, the quality of the acceleration estimation decreases with the random configuration. Tables 1 and 2 show the standard deviation of the acceleration estimation errors.

Table 1. Standard deviation of the acceleration estimation errors, for the symmetric and random 27 IMUs configurations within the 20 mm edges cube.

IMUs Distribution	Axis	Locations Considered (m/s^2)			Locations Unconsidered (m/s^2)		
		Centralized	Distributed	Baseline	Centralized	Distributed	Baseline
Symmetric	x axis	0.0136	0.0146	0.0150	0.0136	0.0094	0.0150
	y axis	0.0139	0.0158	0.0149	0.0139	0.0098	0.0149
	z axis	0.0139	0.0160	0.0149	0.0139	0.0097	0.0149
Random	x axis	0.0142	0.0150	0.0171	0.0158	0.0124	0.0169
	y axis	0.0142	0.0174	0.0173	16.01	16.11	16.01
	z axis	0.0142	0.0173	0.0174	22.93	23.08	22.93

Table 2. Standard deviation of the acceleration estimation errors, for the symmetric and random 27 IMUs configurations within the 200 mm edges cube.

IMUs Distribution	Axis	Locations Considered (m/s^2)			Locations Unconsidered (m/s^2)		
		Centralized	Distributed	Baseline	Centralized	Distributed	Baseline
Symmetric	x axis	0.0135	0.0279	0.0148	0.0135	0.0092	0.0148
	y axis	0.0137	0.0227	0.0148	0.0137	0.0096	0.0148
	z axis	0.0138	0.0231	0.0148	0.0138	0.0096	0.0148
Random	x axis	0.0151	0.0377	0.0876	0.0839	0.0833	0.0842
	y axis	0.0144	0.0416	0.0910	160.14	161.12	160.16
	z axis	0.0143	0.0469	0.0914	229.29	230.75	229.32

As expected, without symmetry in the measured fictitious accelerations, the three algorithms not considering sensors location provide completely unusable results. On their part, the algorithms considering sensors location are differently affected by the random configuration. For the three architectures, the performance degradation is larger for the 200 mm edges cube than for the 20 mm one. Also, for the 200 mm edges cube, the degradation is much larger for the baseline architecture than for the two others. The knowledge of the sensor noises characteristics, which is introduced into the Kalman filters, helps at conserving the quality of the estimation with the centralized and distributed

architectures. When the locations are considered, the main reason to explain the performances degradation due to random configuration is the sensors bias estimation. The centralized architecture provides a better estimation of the biases (Section 4.3) than the distributed architecture. It is, therefore, less affected by the random configuration. Both architectures use the redundancy and apparent observability differently; the centralized architecture uses it to improve the estimation of the biases, while the distributed architecture uses it to improve the acceleration and attitude rate estimations.

4.1.2. Relative Distance between the IMUs

The distance between 27 IMUs is varied from 1 mm to 100 mm. A symmetric 27 IMUs configuration, a cube with nine IMUs evenly distributed on each face and one IMU in the middle of the cube, at the centre of mass, is selected (Figure 1). This configuration creates symmetries in the sensed fictitious accelerations.

Figure 2 presents the resulting standard deviation of the acceleration estimation errors and Figure 3 shows the standard deviation of the attitude rate estimation errors. Those results are obtained with a Monte-Carlo simulation composed of 350 runs for each sensor separation. For each run covering the full projectile trajectory, the standard deviation of the estimation errors is computed. The resulting standard deviations are then averaged over the 350 runs, and the result presented as a function of the sensors separation.

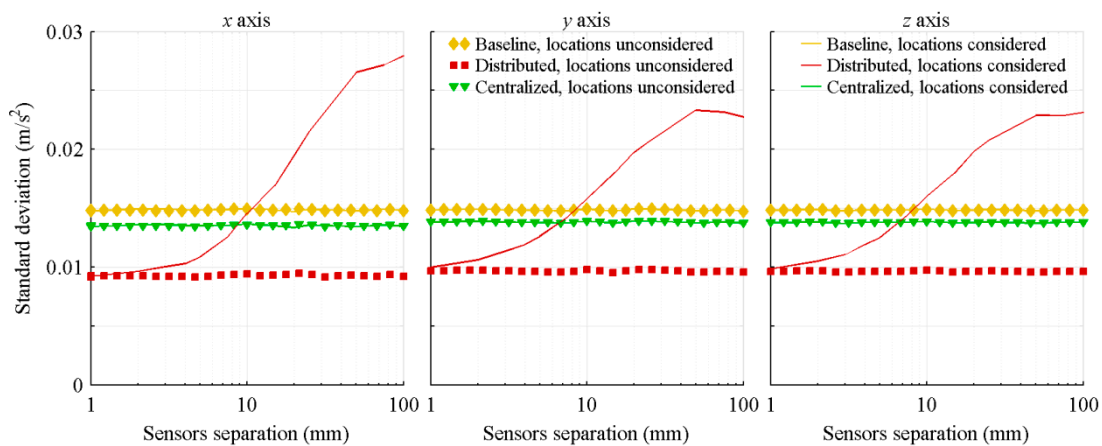


Figure 2. Standard deviation of the acceleration estimation errors, with respect to the distance between the IMUs.

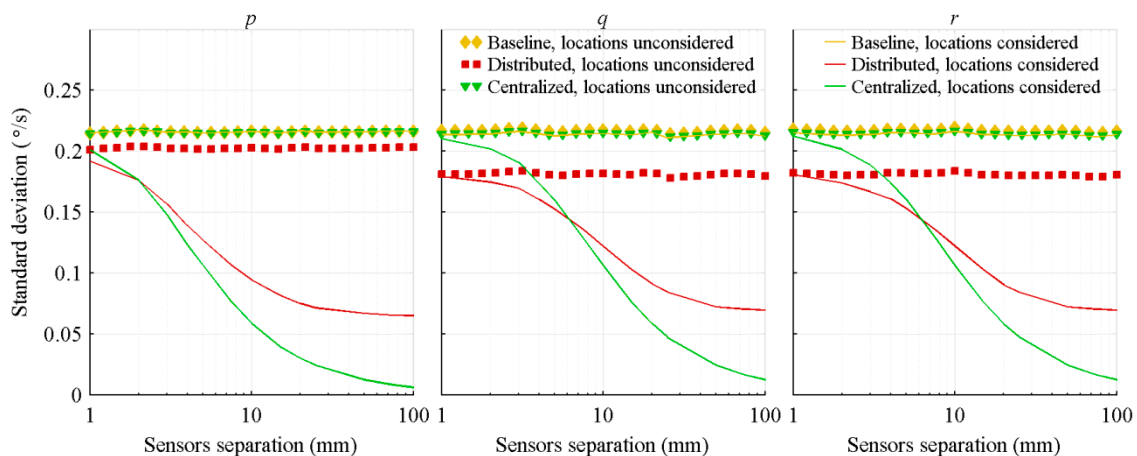


Figure 3. Standard deviation of the attitude rate estimation errors, with respect to the distance between the IMUs.

As a symmetric sensors configuration is used, the algorithms not considering sensors location are still very efficient. For the acceleration estimation with the centralized and baseline architectures, there is no gain at considering the relative locations of the sensors in the algorithms. The fictitious accelerations sensed by one sensor are always cancelled by the corresponding symmetrical sensor. For the distributed architecture, there is even a loss in efficiency when the sensors location is considered in the algorithm. In this architecture, the acceleration of each IMU is estimated prior to the averaging. Therefore, portions of the fictitious accelerations are considered as real accelerations and are not canceled by the sensors symmetry.

Considering the sensors location in the algorithms significantly improves the precision of the attitude rate estimation. By taking into account the sensors location, the accelerometers are combined with the gyroscopes in the attitude rate estimation. When the sensors separation grows, the fictitious accelerations increase and are more easily discriminated over the accelerations of the sensor centre of mass and noises. The performance improvement is therefore more important for larger separation. Also, the centralized architecture better exploits those fictitious accelerations than the distributed architecture. The first one considers the sensors configuration as a whole, and therefore, simultaneously estimates the fictitious accelerations sensed by all accelerometers. On its part, the distributed architecture treats each IMU separately and does not exploit the sensors configuration as a whole. It only uses the location of each IMU with respect to the centre of mass. Therefore, portions of the fictitious accelerations are considered as real accelerations. The worse results of the distributed architecture, in comparison to the centralized architecture ones, are emphasized by its poor biases estimation (Section 4.3).

Also, due to the nature of the projectile trajectory and to the Kalman filters tuning, the three projectile axes show similar performance. The motions of the projectile along its y and z axes are not significantly different, and are in the same order of magnitude as its motion along its x axis (linear deceleration). The only parameter which is significantly different than the others is the projectile x axis spin rate which decreases rapidly. Therefore, a larger value is given to the variance of the projectile spin rate (σ_p^2). For the algorithms not considering sensors location, the improvements of the centralized and distributed architectures in comparison to the baseline one are slightly less important for this parameter (left graphic of Figure 3). However, for the algorithms considering sensors location, the improvements are more significant for this parameter than for the others. The nominal spin rate of the projectile being much larger than the two other rates, the fictitious accelerations due to this rate are larger, and therefore, are more easily distinguishable.

4.1.3. Number of Near Symmetrically Located IMUs

As demonstrated in Section 4.1.2, the symmetry of the sensors significantly influences the estimation performance. Therefore, in this analysis, the locations are selected in order to have a close to symmetric distribution for all numbers of IMUs. In this way, the algorithms not considering sensors location can be applied and provide relatively good performance.

Also, a 10 mm separation between the sensors is selected. For close to symmetric distribution, this distance is the larger one allowing the packaging of all sensors within a projectile fuze. Furthermore, based on Section 4.1.2, this distance still provides a significant gain in the attitude rate estimation.

A Monte-Carlo simulation composed of 350 runs is then executed for each IMU number, which is varied from 2 to 60 by steps of 2. For each run, covering the full projectile trajectory, the standard deviation of the estimation errors is computed. The resulting standard deviations are then averaged over the 350 runs, and the result presented as a function of the number of IMUs in Figures 4 and 5.

As expected, a trend coherent with the signal averaging theory, a diminution of the standard deviation proportional to $\frac{1}{\sqrt{n}}$, is obtained for all cases. For the algorithms not considering sensors location, the distributed architecture provides the best performance. Because of the local filtering, the mean of the master filter is computed on data having a smaller standard deviation than the raw measurements, resulting in a less noisy final estimation for the distributed architecture than for the baseline architecture. On its part, because of the covariance matrices tuning which gives equal weight

to all measurements, the centralized architecture is, when the sensors location is not considered, similar to directly computing the mean of the measurements. The small performance improvement, in comparison to the baseline architecture, comes from the a priori knowledge of the trajectory and noises introduced into the Kalman filter. The two Kalman filterbased architectures, therefore, use the redundancy differently: the centralized one improves the biases estimation (Section 4.3), while the distributed one improves the parameter estimations.

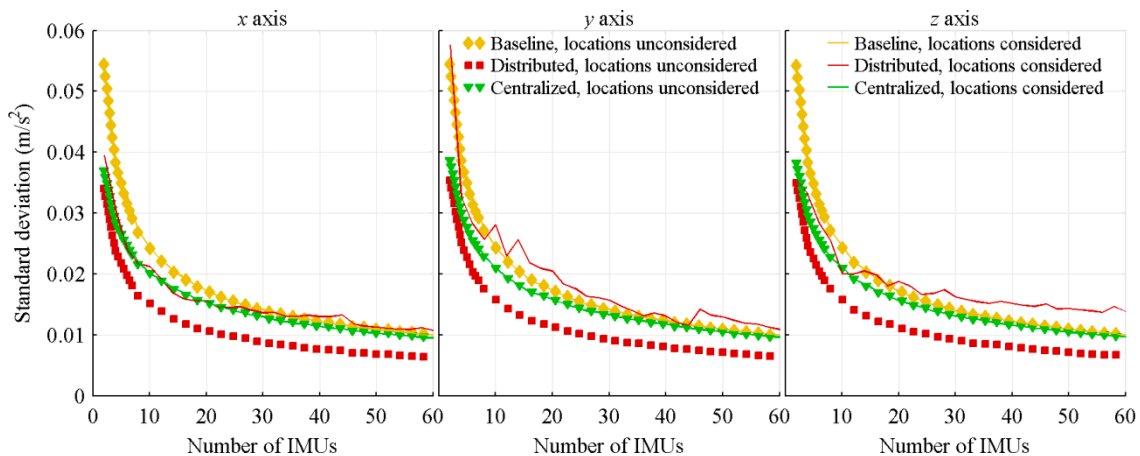


Figure 4. Standard deviation of the acceleration estimation errors, with respect to the number of IMUs.

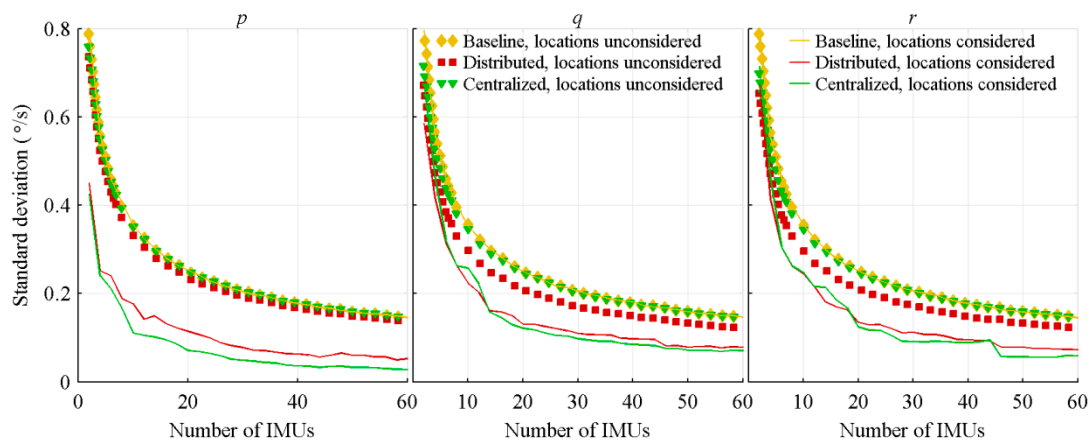


Figure 5. Standard deviation of the attitude rate estimation errors, with respect to the number of IMUs.

However, as in Section 4.1.1, due to how it handles the fictitious accelerations, the centralized architecture with its algorithm considering sensors location is the best option for nearly all tested number of IMUs. The light geometric dissymmetry for some numbers of sensors, and the non-linear variation, with respect to the number of IMUs, of the distance between the centre of mass of the IMUs configuration and the farthest IMU, explain the non-smooth behaviour of the distributed architecture acceleration estimation and of the centralized and distributed architectures attitude rate estimation.

As in Section 4.1.1, the algorithms considering sensors location are, for the attitude rate estimation, better than those not considering them, but the distributed architecture acceleration estimation is worse when the sensors location is considered. Also, the acceleration estimation errors of the centralized and baseline architectures are not significantly modified by the inclusion of the sensors location within the algorithms. As in Section 4.1.1, because of the large nominal value of the spin rate, the estimation of this parameter differs from the others; the centralized and distributed architectures with their algorithms not considering sensors location are slightly less efficient for the spin rate axis than for the

other axes, the opposite being observed for those architectures when the sensors location is considered in the algorithms.

All previous results were obtained with fully uncorrelated bias instabilities. However, MEMS are expected to have some correlations between them. Therefore, a second series of tests, with fully correlated biases is done. For these tests, the same bias is applied to all sensors generating measurements for the same parameter. There are no correlations between the axes and, as it is nearly impossible to know the sensors' correlation in a real system, the knowledge of the correlation is not included in the Kalman filter covariance matrices.

The following graphics compare the attitude rate (Figure 6) and acceleration (Figure 7) estimation errors of the correlated biases tests to the previous uncorrelated ones. Obviously, the biases of the sensors measuring the same parameter are not expected to be fully correlated nor fully uncorrelated. However, comparing the two sets of results provides bounds for the achievable performance. In order to keep the graphics readable, the results of the algorithms considering sensors location were included only. For correlated biases, the three algorithms not considering sensors location provide estimations nearly identical to those of the baseline algorithm considering sensors location.

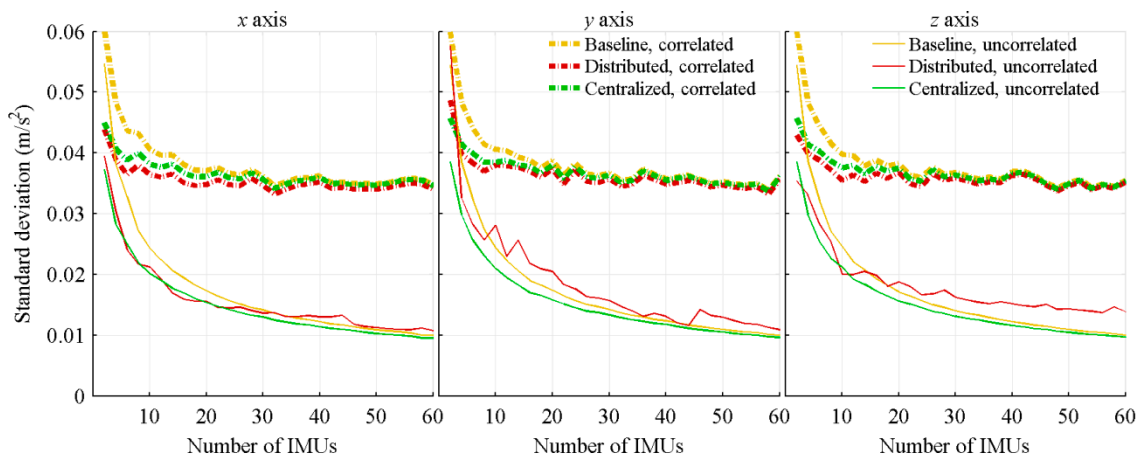


Figure 6. Standard deviation of the acceleration estimation errors, with correlated and uncorrelated sensors bias, and function of the number of IMUs.

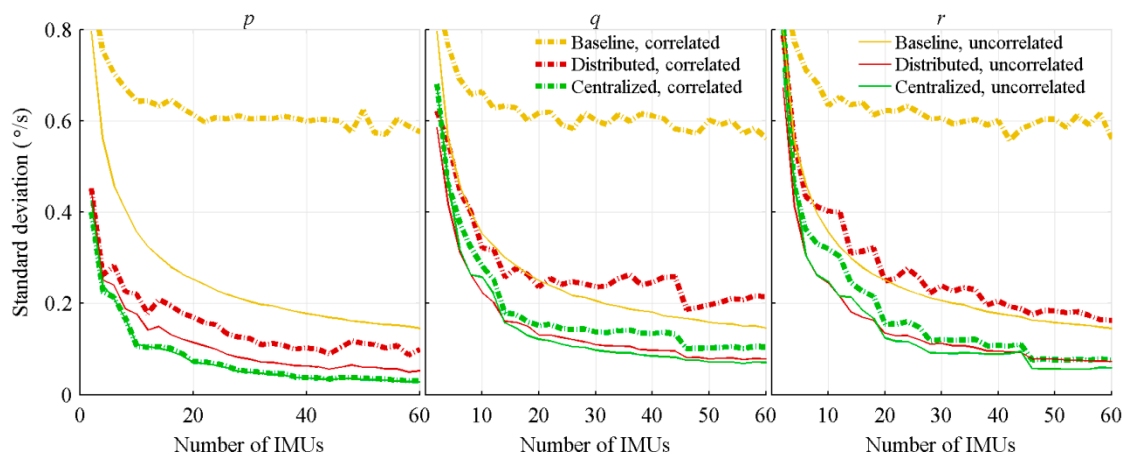


Figure 7. Standard deviation of the attitude rate estimation errors, with correlated and uncorrelated sensors bias, and function of the number of IMUs.

When the sensors location is considered, the accelerometers provide supplementary information on the attitude rate, allowing the distinction between the biases and vehicle motions. This is the

main explanation of the improvements, with respect to the number of sensors, obtained with fully correlated biases. On their parts, the three architectures' acceleration estimation and the baseline attitude rate estimate provide, as mentioned in the literature [20], small improvements with respect to the number of considered IMUs. The biases cannot be clearly distinguished from the vehicle motions, and increasing the number of IMUs does not fix the issue.

Up to this point, the analysis is focusing on the precision of the estimation. The second important characteristic of the quality of an estimation is its accuracy, characterized by the mean of the estimation errors. For the tested cases, the mean is always near 0 and does not show a specific trend with respect to the number of IMUs.

The third studied characteristic of the algorithms is the relative computational time, which is presented in Figure 8 as a function of the number of IMUs. As the sensors' bias correlation does not affect the computational time, only the times of the fully uncorrelated biases tests are presented. The reference computational time is the computational time of the baseline architecture with its algorithm not considering sensors location in the case of two IMUs. Table 3 summarizes the operation that requires the most computational time for each architecture and the number of times this operation must be performed.

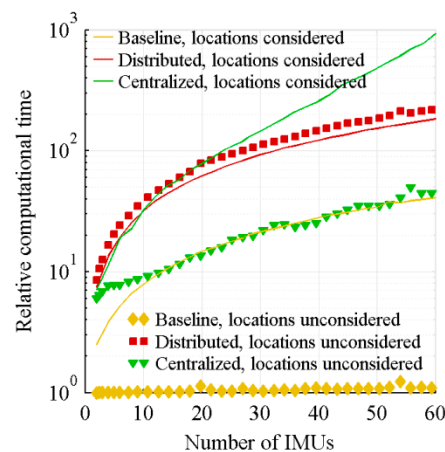


Figure 8. Relative computational time.

Table 3. Most time consuming operation for each architecture.

Architecture	Locations Considered	Locations Unconsidered
Centralized	Invert a $6n \times 6n$ matrix	Invert $6n \times n$ matrices
Distributed	Invert $n \times 6$ matrices	Compute $6n$ divisions
Baseline	Compute $3n$ cross-products	Mean computations

For the baseline architecture, the data fusion is implemented through, quickly performed, basic mathematical operations and the computation time variation as a function of the number of IMUs is marginal for the algorithm not considering sensors location. However, for the algorithm considering sensors location, the estimation of the angular acceleration which requires n cross-products is more time consuming and produces an increase of the computational time proportional to the number of sensors.

For the distributed architecture, the computational time increments come from the time required to compute the added local filters. For the algorithm not considering sensors location, each local filter has to compute six divisions, while for the algorithm considering sensors location, it must invert a 6×6 square matrix. Both operations are similar in term of computational time, the small difference being rather due to the larger amount of data handling operations for the algorithm not considering sensors location.

For the centralized architecture, the computational time is driven by the inversion of the Kalman filter covariance matrix. When the sensors location is considered, the addition of an IMU adds six rows and six columns to the matrix to be inverted. However, when the sensors location is not considered, it rather adds one row and one column to each of the six covariance matrices. For a relatively small symmetric matrix, the inversion is quickly performed. Therefore, among the algorithms not considering sensors location, the centralized architecture, which requires less handling operations, is faster than the distributed one. However, for large matrix, typical of the centralized architecture with its algorithm considering sensors location, the computational time is much longer.

The presented data fusion algorithms do not provide attitude, velocity and position estimations, which are provided by a standard INS. Also, without GPS receiver, there is no corrections possibility on these estimations. The estimated attitude rate is therefore directly integrated to estimate the attitude. On its part, the estimated acceleration, which is measured in the body frame, is transposed, by a rotation matrix, in the North-East-Down (NED) frame and integrated, in this frame, to estimate the position and velocity.

The standard deviation of the position errors is shown in Figure 9. As a comparison point, the typical precision of a military grade GPS receiver is included in the graphics.

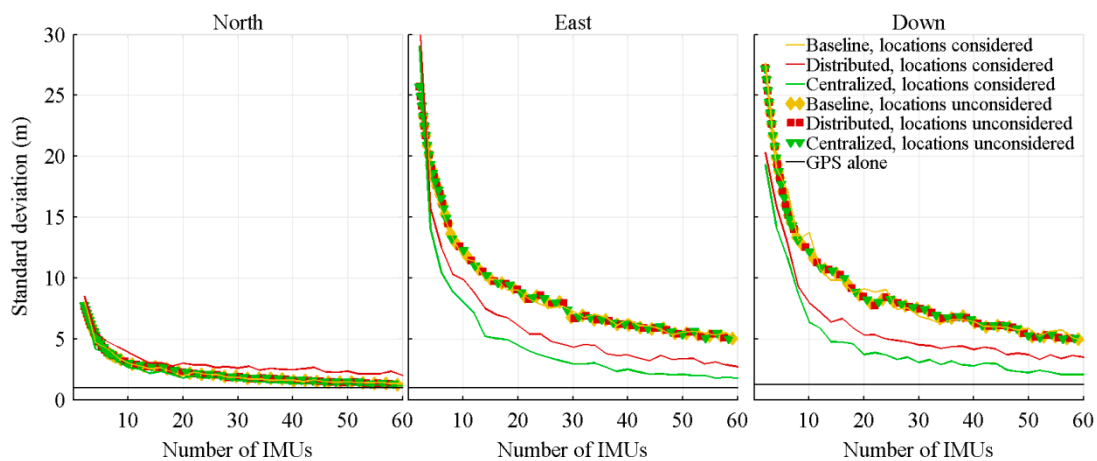


Figure 9. Standard deviation of the position errors when the projectile hits the ground, with respect to the number of IMUs.

Even if the precision of the acceleration and attitude rate estimations differ between the algorithms, the standard deviation of the position errors when the projectile hits the ground is not significantly different between the algorithms not considering sensors location, the gaps being not even visible in the graphics. The distributed architecture is better than the two others by, approximately, 0.01 m.

However, for the algorithms considering sensors location, the centralized and distributed architectures offer a significant gain over the baseline architecture, the improvements coming from the better attitude rate estimation. The accelerations are measured in the body frame, while the positions are expressed in the NED coordinates. To migrate from the body frame to the NED frame, the accelerations must be rotated. The rotation is performed by a rotation matrix which is directly affected by the attitude rate estimation.

4.2. Fault Detection

The efficiency of the algorithm proposed for FDI is tested on a specific number of IMUs and a fixed number of faults. The symmetric 27 IMUs configuration of Section 4.1.2, with three faults for the accelerometers and three faults for the gyroscopes, is used to assess the FDI performance of the three architectures equipped with the proposed FDI algorithm. A single form of faulty sensor, a sensor producing a measurement related to the vehicle motion, but with a larger random walk than the sensor

typical one, is studied. The faulty sensor, the time of the fault and the amplitude of the faulty random walk are randomly set prior to each simulation. The amplitude of the random walk of the faulty sensor is, at least, 1.75 times that of a healthy sensor, and the fault occurs, at least, 1 s prior to the end of the simulation.

In this section, only the algorithms considering sensors location are studied. The algorithms not considering sensors location quickly consider all accelerometers as faulty, because they assimilate the fictitious accelerations to the sensors bias. Figures 10 and 11, respectively, show the mean numbers of correctly detected faulty sensors and of false warnings, based on Monte-Carlo simulations of 100 runs each.

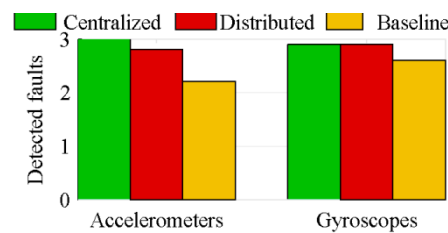


Figure 10. Detected faults for the algorithms considering sensors location.

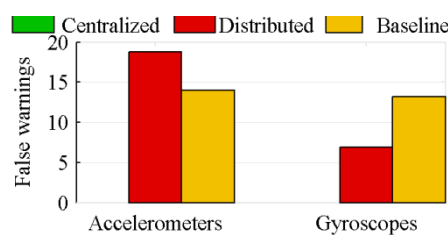


Figure 11. False warnings for the algorithms considering sensors location.

Because of its inability to correctly estimate the sensors bias, the distributed architecture generates a lot of false warnings, while the centralized architecture correctly identifies nearly all faulty sensors. The effect of this poor biases estimation is shown in Figure 12. This figure presents the 4σ thresholds of the centralized and distributed architectures over the measurements of a healthy sensor, when 27 IMUs are used. Because of its drift, at 8.9 s, this healthy sensor is considered faulty by the distributed architecture. The ideal threshold, added to the graphic, is the threshold computed on the nominal biased value. The latter is the measured value without noise.

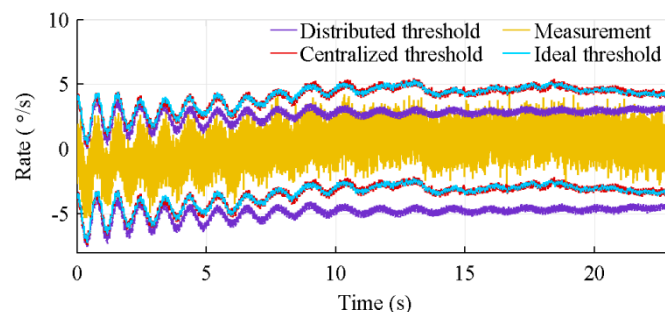


Figure 12. Typical 4σ thresholds of the centralized and distributed architectures with the algorithms considering sensors location, for 27 IMUs.

The bias estimation also explains why there are more false warnings for the accelerometers than for the gyroscopes. Because of the fictitious forces, the measured acceleration magnitude is much

larger than the measured attitude rate one. The distributed architecture has difficulty to estimate the sensors bias in presence large measured values (Section 4.3).

The proposed FDI algorithm is therefore dependent of the number of IMUs. With more IMUs, the standard deviation of the residual decreases and the bias estimation improves. A smaller standard deviation smoothes the threshold bounds, while a bias estimation improvement tightens them. Figure 13 shows, for 8 and 64 IMUs, the threshold of the centralized architecture and the detection time of a faulty sensor. Because of a better threshold estimation, the fault is detected nearly 6 s earlier with 64 IMUs.

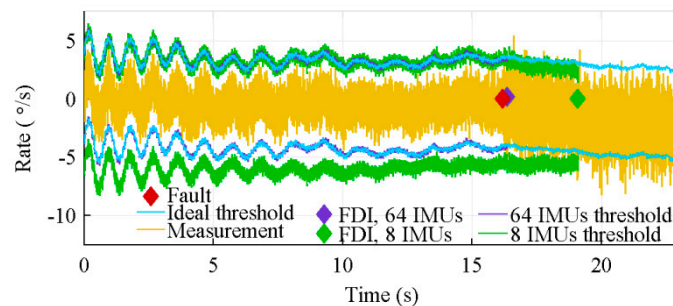


Figure 13. Typical 4σ threshold of the centralized architecture with the algorithm considering sensors location, for 8 and 64 IMUs.

4.3. Bias Estimation

Most of the results obtained previously are highly influenced by the inclusion of the sensors bias estimation within the centralized and distributed architectures. An analysis of this bias estimation is therefore done.

Both, the distributed and centralized architectures, have an unobservable subspace. However, they both show the typical behaviour of a fully observable system. Notably, they are able to come back to the true value after a significant outlier. This apparent observability comes from the correlation between the measurements, which is used to extract supplementary information for the sensors bias estimation. The centralized architecture is however better at estimating those biases, as shown in Figure 14, which shows the estimation of a bias on a gyroscope measuring the spin rate in a 8 and a 64 sensors configuration, the sensors location being considered. Furthermore, also visible on Figure 14, with more sensors, which favour the extraction of supplementary information, the biases estimation is improved.

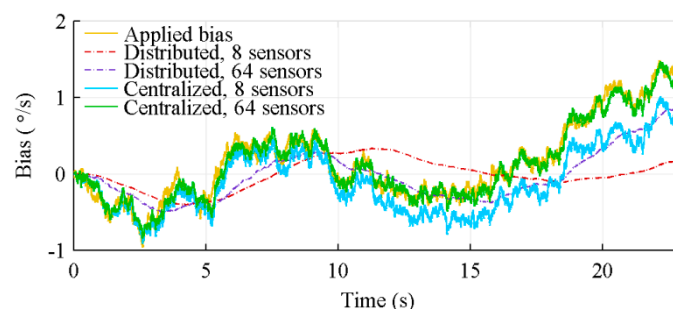


Figure 14. Bias estimation of gyroscopes measuring p , for 8 and 64 IMUs, with algorithms considering sensors location.

Figure 15 illustrates the difficulty of the distributed architecture, due to the a priori knowledge of the trajectory (sensor movements) introduced into the Kalman filter state noise covariance matrix, to estimate the sensors bias when the expected measurement variation is large. In this figure, the same

bias is applied on gyroscopes measuring p (large value) and q (small value). This figure shows that the centralized architecture bias estimation is marginally affected by the amplitude of the measurements, while the distributed architecture one is significantly better for q .

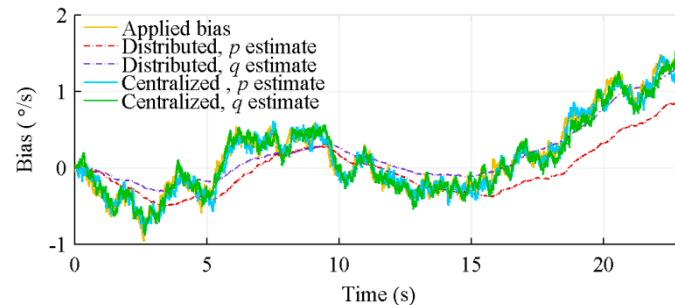


Figure 15. Bias estimation of gyroscopes measuring p and q , for 64 IMUs, with algorithms considering sensors location.

5. Conclusions

This work has analysed two data fusion architectures, a centralized architecture and a distributed architecture, to treat the measurements of a large amount of orthogonal redundant inertial measurement units. For each architecture, an algorithm considering the location of the sensors and another one not considering it have been proposed. In the centralized architecture, all sensor measurements are simultaneously processed by a single Kalman filter. In the distributed architecture, the measurements of each sensor are filtered by a local Kalman filter before being sent to a master data fusion filter which computes the final estimate. A baseline architecture, which directly computes the mean of the measurements, is used as a point of comparison. As this work focused on Micro-Electronic-Mechanical Systems, whose one of their main weaknesses is their large bias instability, the implemented Kalman filters were designed to estimate the sensors bias.

The Monte-Carlo simulations demonstrated that the centralized architecture with its algorithm considering sensors location provides the best performance. Its estimations of the accelerations, attitude rates and sensors bias are the most precise and accurate. It is also less affected when the IMUs are randomly located and its acceleration estimation does not degrade when the distance between the IMUs is increased. Additionally, it provides better fault detection capabilities, detecting nearly all faults and generating few false warnings. However, its better performance comes at the expense of a much longer computational time. The distributed architecture with its algorithm considering sensors location provides, at a much lower computational time, good estimations of the accelerations and attitude rates. However, the quality of these estimations degrades when the IMUs are located randomly, and its fault detection capabilities are very limited. The centralized and distributed architectures with their algorithms not considering sensors location remain viable for nearly symmetric locations, but they are completely unusable when the IMUs are located randomly, and for fault detection. With correlated sensors bias, only the centralized and distributed architectures with their algorithms considering sensors location are able to obtain a significant improvement, with respect to the number of IMUs, of the attitude rate estimation. Also, in presence of sensors bias correlation, no matter which architecture is used, the quality of the acceleration estimation is barely modified by the number of sensors.

The results and analyses of this paper were obtained under perfect conditions for the filtering algorithms, which means that the latter were always set with perfect knowledge of the sensors' characteristics and their locations. This study could be extended to cases considering an approximate knowledge of this information. Also, a more in-depth analysis, involving real flight data with a specific number of IMUs, would be done to validate the simulation results.

Author Contributions: E.G. and A.V. conceived and designed the experiments. A.V. and Y.B. developed the algorithms. A.V. performed the tests. E.G., A.V., and Y.B. analyzed the results and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nilsson, J.-O.; Skog, I. Inertial sensors arrays—A literature review. In Proceedings of the 2016 European Navigation Conference, Helsinki, Finland, 30 May–2 June 2016. [\[CrossRef\]](#)
2. Bancroft, J.; Lachapelle, G. Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors* **2011**, *11*, 6771–6798. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Pesja, A. Optimum Skewed Redundant Inertial Navigators. *AIAA J.* **1974**, *12*, 899–902. [\[CrossRef\]](#)
4. Waegli, A.; Guerrier, S.; Skaloud, J. Redundant MEMS-IMU integrated with GPS for Performance Assessment in Sports. In Proceedings of the IEEE/ION PLANS 2008, Monterey, CA, USA, 5–8 May 2008. [\[CrossRef\]](#)
5. Shim, D.-S.; Yang, C.-K. Optimal Configuration of Redundant Inertial Sensors for Navigation and FDI Performance. *Sensors* **2010**, *10*, 6497–6512. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Giroux, R. Capteurs bas de Gamme et Systèmes de Navigation Inertielle: Nouveaux Paradigmes D'application, Ph.D. Thesis, École de Technologie Supérieure, Montréal, QC, Canada, 17 September 2004.
7. Levy, L.J. Suboptimality of Cascaded and Federated Kalman Filters. In Proceedings of the 52nd Annual Meeting of the Institute of Navigation, Cambridge, MA, USA, 19–21 June 1996.
8. Allerton, D.; Jia, H. A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems. *J. Navig.* **2005**, *58*, 405–417. [\[CrossRef\]](#)
9. Lawrence, P.; Berarducci, M. Comparison of federated and centralized Kalman filters with fault detection considerations. In Proceedings of the IEEE PLANS 1994, Las Vegas, NV, USA, 11–15 April 1994. [\[CrossRef\]](#)
10. Sukkariéh, S.; Gibbens, P.; Groscholsky, B.; Willis, K.; Durrant-Whyte, H.F. A Low-Cost, Redundant Inertial Measurement Unit for Unmanned Air Vehicles. *Int. J. Robot. Res.* **2000**, *19*, 1089–1103. [\[CrossRef\]](#)
11. Miller, M.M.; Soloviev, A.; Uijt de Haag, M.; Veth, M.; Raquet, J.; Klausutis, T.J.; Touma, J.E. *Navigation in GPS Denied Environments: Feature-Aided Inertial Systems*; Air Force Research Laboratory: Eglin AFB, FL, USA, 2010.
12. Bachrach, A.; Prentice, S.; He, R.; Roy, N. RANGE—Robust Autonomous Navigation in GPS-denied. *J. Field Robot.* **2011**, *28*, 644–666. [\[CrossRef\]](#)
13. Wu, A.D.; Johnson, E.N.; Kaess, M.; Dellaert, F.; Chowdhary, G. Autonomous Flight in GPS-Denied Environmnets Using Monocular Vision and Inertial Sensors. *J. Aerosp. Inf. Syst.* **2013**, *10*, 172–186. [\[CrossRef\]](#)
14. Grigorie, T.L.; Botez, R.M.; Sandu, D.G.; Grigorie, O. Experimental Testing of Data Fusion Algorithm for Miniaturized Inertial Sensors in Redundant Configurations. In Proceedings of the 2014 International Conference on Mathematical Methods, Mathematical Models and Simulation in Science and Engineering, Interlaken, Switzerland, 22–24 February 2014.
15. Chen, T. Design and Analysis of a Fault-Tolerant Coplanar Gyro-Free Inertial Measurement Unit. *J. Microelectromechan. Syst.* **2008**, *17*, 201–212. [\[CrossRef\]](#)
16. Skog, I.; Nilsson, J.-O.; Händel, P.; Nehorai, A. Inertial Sensor Arrays, Maximum Likelihood, and Cramér–Rao Bound. *IEEE Trans. Signal Process.* **2016**, *64*, 4218–4227. [\[CrossRef\]](#)
17. Sabatini, A. Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation. *Sensors* **2011**, *11*, 9182–9206. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Tereshkov, V. A Simple Observer for Gyro and Accelerometer Biases in Land Navigation Systems. *J. Navig.* **2015**, *68*, 635–645. [\[CrossRef\]](#)
19. Noureldin, A.; Karamat, T.; Eberts, M.; El-Sahfie, A. Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications. *IEEE Trans. Veh. Technol.* **2009**, *58*, 1077–1096. [\[CrossRef\]](#)
20. Jiang, C.; Xue, L.; Chang, H.; Yuan, G.; Yuan, W. Signal Processing of MEMS Gyroscope Arrays to Improve Accuracy Using a 1st Order Markov for Rate Signal Modeling. *Sensors* **2012**, *12*, 1720–1737. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Schwaab, M.; Reginya, S.; Sikora, A.; Abramov, E. Measurement analysis of multiple MEMS sensor array. In Proceedings of the 24th Saint Petersburg International Conference on Integrated Navigation Systems, St. Petersburg, Russia, 29–31 May 2017. [\[CrossRef\]](#)
22. Woodman, O. *An Introduction to Inertial Navigation*; University of Cambridge: Cambridge, MA, USA, 2007.

23. Zipfel, P.H. *Modeling and Simulation of Aerospace Vehicle Dynamics*, 2nd ed.; AIAA Inc.: Reston, VA, USA, 2000; ISBN 1-56347-875-7.
24. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2011; ISBN 1-58488-391-X.
25. Ray, A.; Phoba, S. Calibration and estimation of redundant signals. In Proceedings of the 2002 American Control Conference, Anchorage, AK, USA, 8–10 May 2002. [[CrossRef](#)]



© 2018 Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).