

Generic multisensor multitarget bias estimation architecture

J.A. Besada G. de Miguel A.M. Bernardos J.R. Casar

Abstract: Current bias estimation algorithms for air traffic control (ATC) surveillance are focused on radar sensors, but the integration of new sensors (especially automatic dependent surveillance-broadcast and wide area multilateration) demands the extension of traditional procedures. This study describes a generic architecture for bias estimation applicable to multisensor multitarget surveillance systems. It consists on first performing bias estimations using measurements from each target, of a subset of sensors, assumed to be reliable, forming track bias estimations. All track bias estimations are combined to obtain, for each of those sensors, the corresponding sensor bias. Then, sensor bias terms are corrected, to subsequently calculate the target or sensor–target pair specific biases. Once these target-specific biases are corrected, the process is repeated recursively for other sets of less reliable sensors, assuming bias corrected measures from previous iterations are unbiased. This study describes the architecture and outlines the methodology for the estimation and the bias estimation design processes. Then the approach is validated through simulation, and compared with previous methods in the literature. Finally, the study describes the application of the methodology to the design of the bias estimation procedures for a modern ATC surveillance application, specifically for off-line assessment of ATC surveillance performance.

1 Introduction

Multisensor multitarget tracking systems are the basis of modern air traffic control (ATC) systems. They rely on highly accurate tracking filters (such as IMM filters [1]), which exploit all the available sensor measurements, enabling extremely fast manoeuvre detection and manoeuvre adapted filtering. These filters manoeuvre detectors are based on the assessment of the discrepancy between actual measures and track state prediction, derived from previous measures. In order to be computationally efficient, zero-mean Gaussian measurement error statistics are usually assumed, and therefore any mismatch between this model and the real measurements actually leads to manoeuvre detection problems, reducing the quality of the tracking results.

Owing to this problem, much effort has been devoted in the last years to the definition of bias estimation procedures for multisensor multitarget tracking systems (e.g. [2–12]). The basic idea is estimating every bias terms in the measurements potentially causing consistency mismatch, and removing them from raw measures, providing the tracking filters with bias-corrected (mostly unbiased) measures.

These efforts have been mainly focused on radar bias estimation and correction, which were the most widely used sensors for these applications. Nowadays, most surveillance systems rely on the use of different sensors with complementary behaviour. It allows a number of benefits (high accuracy, extended coverage, enhancements to systematic error estimation, correction, etc.) and brings new

challenges to the fusion process: it requires a robust strategy that considers the specific characteristics of all data sources and checks their consistency before being fused. Thus, in particular, the means for bias estimation must be extended.

Bias terms can be divided in several subtypes:

- Sensor-related bias that have a same value independently of the target.
- Target-related bias, equal for every sensor of the same type, independently of the sensor.
- Sensor–target pair related bias, different for each sensor–target pair.

Most of the bias estimation literature focus on the derivation of high-quality estimators to obtain sensor-related biases, and tend to neglect the importance of target-related biases. Blom *et al.* [3] is a notable exception to this: it defines two kinds of bias terms (‘micro’ and ‘macro’) and it derives a complete bias estimation algorithm and architecture to jointly estimate both terms and target kinematic state. In this paper, we provide a comprehensive approach to the derivation of a bias estimation and correction subsystem which:

- Enables easy extension to new bias terms and sensors. It is not only designed for radar sensors, but it can deal with other surveillance sensors.
- Gives adequate importance to target-related and sensor–target pair-related bias, which can now be estimated and corrected because of the availability of better information sources.

- Provides the basis-to-potential improvements related to the assessment and management of integrity.

The methods proposed in this paper are based on the definition of error models for the sensors, and define an architectural framework for bias estimation that might be extended to other applications such as airport surface surveillance, defence surveillance systems, coastal surveillance etc. The final sections of the paper describe the application of the proposed approach to the derivation of a bias estimation system included in a key module of the new version of Eurocontrol's tool for the assessment of ATC surveillance called surveillance analysis support system for centres (SASS-C) [13]. This module, called opportunity trajectory reconstruction (OTR, or SASS-C v7/OTR), is a batch process where all the measurements from the available sensors are processed to obtain smoothed trajectories for all targets in the area of interest. It requires accurate measurement-to-reconstructed trajectory association, bias estimation and correction to align different sensor measurements, and adaptive multisensor smoothing to obtain the final interpolated trajectory. Previous SASS-C versions had simplified bias estimation algorithms, focused in radar sensors. In fact, OTR was aimed at completely redefining SASS-C reconstruction algorithms to deal with all new sensors of interest.

The paper is structured as follows. It starts with the definition of an abstracted error model for the sensors' measurements (Section 2), and then describes and justifies the bias estimation architecture (Section 3), providing mathematical expressions for the estimation at the different stages of the architecture. Most of the used estimator definitions can be derived from filters described in any estimation theory book or tutorial paper [14, 15]; what is innovative here is the derivation of the complete architecture and the optimised design for bias estimation process. Section 4 includes some results in a simplified example of application trying to show how our proposal effectively manages the different types of errors, and comparing its results with others in the literature.

Then, it describes the application of the methodology for ATC reconstruction (Section 5), including the definition of the error models for the sensors of interest, and describes its details in this implementation. The feasibility of the proposed approach is demonstrated through a set of results from simulated and real scenarios (Section 6).

2 Sensor error models

In general, for a given sensor, each measurement may be modelled as a function of the actual (ideal) target state, systematic errors (bias) and random errors (noise). In our paper, sensors will only provide position measurements, although it could be extended to other kinematic data. So, being (x_k, y_k) the measured position projected into a common horizontal plane, it can be modelled as

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \Delta\theta_s, \Delta\theta_t, \Delta\theta_{s-t}, n_s(k) \right) \quad (1)$$

where

- $(x_{\text{id}}(k), y_{\text{id}}(k))$ is the ideal position of the target for the k th measurement.

- $(\Delta\theta_s, \Delta\theta_t, \Delta\theta_{s-t})$ are the biases vectors including, respectively, sensor biases, target biases and biases related to sensor-target pair.
- $n_s(k)$ is a vector containing the noise errors corrupting the k th measurement.

Quite often sensor design permits performing an accurate first-order approximation of (1) in the vicinity of the target position [2, 5], as in

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \end{bmatrix} &\cong \begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix} + [\mathbf{H}_s(k) \quad \mathbf{H}_t(k) \quad \mathbf{H}_{s-t}(k)] \begin{bmatrix} \Delta\theta_s \\ \Delta\theta_t \\ \Delta\theta_{s-t} \end{bmatrix} \\ &+ \mathbf{G}_s(k)n_s(k) = \begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix} + \mathbf{H}_e(k) \begin{bmatrix} \Delta\theta_s \\ \Delta\theta_t \\ \Delta\theta_{s-t} \end{bmatrix} \\ &+ \mathbf{G}_s(k)n_s(k) \end{aligned} \quad (2)$$

where:

- $\mathbf{H}_\theta(k)$ is the time-changing bias projection matrix for k th measurement. It is a $2 \times N$ matrix, where N is the number of bias terms in error model, and can be decomposed in three submatrices related to each kind of bias (respectively, $\mathbf{H}_s(k)$, $\mathbf{H}_t(k)$, and $\mathbf{H}_{s-t}(k)$).
- $\mathbf{G}_s(k)$ is the time-changing noise projection matrix for k th measurement. It is a $2 \times M$ matrix, where M is the number of noise terms in sensor error model.

The model in (2) is the basis for the bias estimation algorithms described in the paper. It should be noted that some of the bias terms, although assumed constant in our models, are in fact slowly changing in time, at a rate dependent on different measurement aspects (propagation changes, drift of other sensor calibration means as station synchronisation procedures for multilateration etc). A system trying to cope with these terms should define means to either

- Forget sensor bias over time, by smoothing successive bias estimations.
- Include sensor models enabling changing bias, which could demand modifications of the algorithms described here.

3 Bias estimation and correction methodology

The basis of every bias estimation procedure is exploiting the information about the bias terms present in measurements, by cancelling the dependency with the ideal target position. This is done through the comparison of the different measurements of the same target from several sensors, and reducing the noise effect on the estimation by averaging noise samples. It can be accomplished by extrapolation of pairs of measurements to a common time reference and posterior differentiation (see, for instance, [2, 6]), by the procedure described in this paper or by other techniques (e.g. such as definition of pseudo measures from combinations of measures [4, 5, 10] or parallel target state and bias estimation [3, 7, 9]). In general, independently of the method applied, an observation of the bias terms is obtained from each measure or group of measures coming from the involved sensors. This single

observation (or set of observations) does not allow discriminating the different bias terms. The basis of the bias estimation procedures is the availability of a diverse traffic pattern, providing measures with changing bias projection matrices ($H_{\theta}(k)$), and changing sensor combinations in each track so that:

- There are measurements of all the sensors involved.
- Every pair of sensors is connected, from the point of view of the bias estimation, by a track or a set of tracks. A pair of sensors is connected if there is a track with measures from both sensors or if, by a transitive property, there is a set of tracks, which, through a set of intermediate connected sensors, connect them.

There are several problems to be addressed when defining the architecture for bias estimation, mainly related to the observability [14] of each bias term. There are mainly two kinds of observability problems:

1. Impossibility to separate some terms from the same sensor or from several sensors for a certain target, as they provoke the same position offset for the whole trajectory.
2. Impossibility to separate some target or sensor–target biases from sensor biases, because of their additive nature.

The first problem may be addressed, for sensor-related biases, by averaging data from different targets. For target or sensor–target-related biases those problems need to be addressed using *ad hoc* procedures. Meanwhile, the solution of the second problem demands exploiting the inherent difference of target and sensor–target biases for each target, and averaging the terms not related to sensor biases in a controlled manner.

The proposed bias estimation architecture is derived from the previous characterisation of the bias, being oriented to solve these observability problems. Its integration in a complete tracking or reconstruction architecture can be seen in Fig. 1, where the arrows describe data-flow—not execution order, as all those processes may be parallelised for real-time systems. In this paper, perfect previous association is assumed.

As previously stated, the proposed bias estimation and correction architecture uses associated data from several sensors. It may be designed as a recursive operation, as the process may be decomposed in a set of estimation/correction stages, estimating different bias terms and correcting them in a predefined order.

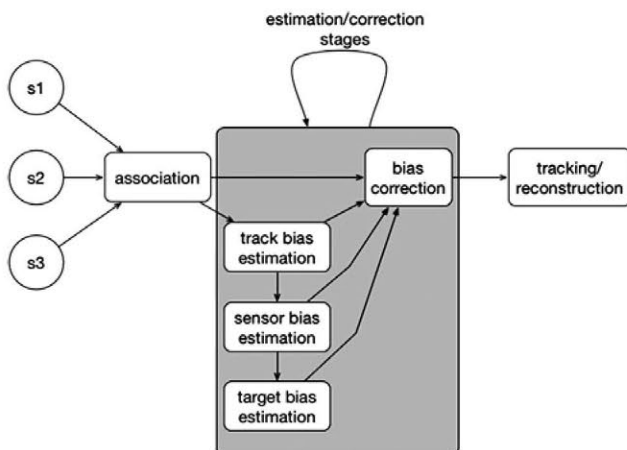


Fig. 1 Bias estimation architecture

There are three kinds of measurements at each estimation stage:

1. Measurements from sensors not to be used at this estimation stage (to be called ‘unusable measures’). They could be used in other stages.
2. Raw measures, containing bias information, from the sensors whose bias terms are to be estimated and corrected at this bias estimation stage (to be called ‘biased measures’).
3. Previously corrected measurements (in previous estimation/correction stages), assumed to be unbiased (to be called ‘corrected measures’).

The three steps conforming each stage are:

1. Track bias estimation: from all the corrected and biased measures from a given target (integrated in a multisensor track), this step obtains an estimation of all the bias terms from all the sensors providing biased measurements to this track. In our approach, only the measurements obtained during constant velocity movement segments will be used. This is needed because of the high sensitivity of the described estimation algorithms to manoeuvres, although other algorithms not demanding this segmentation could be used. In Section 3.1, the general method for this function is described. It should be noted that, in this stage, some of the second kind of observability problems previously described would be addressed by jointly estimating the problematical variables. Subsequent steps to be described next will try to solve those problems. As important as obtaining a vector of bias estimates is obtaining a consistent estimation of its covariance matrix, to be exploited later.
2. Sensor bias estimation: this method integrates every track bias estimator in an efficient manner, to obtain sensor-related biases. It assumes that the previously described bias vectors are independent measurements (as they come from different target measurements) of the sensor bias terms. Those measurements have a covariance matrix, which can be derived from the results of track bias estimation. In Section 3.2, the complete sensor bias derivation procedure is outlined.
3. Target and Sensor–target bias estimation: this is obtained for each target, provided that sensor-related bias was previously corrected. This estimation can be performed either in parallel or almost in parallel with target tracking, especially for real-time surveillance systems. It will be studied in Section 3.3.

3.1 Track bias estimation

The track bias estimation algorithm performs bias estimations on the basis of the measurements feeding a given multisensor track. Therefore it only estimates the terms related to the sensors originating these measurements. The basic algorithm in this paper is based on a Kalman filter with stacked parameters, but the proposed architecture is compatible with other bias estimation algorithms. Kalman state variables are:

1. Target two-dimensional (2D) position.
2. Target two-dimensional (2D) velocity.
3. Bias parameters to be jointly estimated for all the sensors feeding the multisensor track.

Each target is under coverage of a different set of sensors, and therefore the state vector of each Kalman filter is different. Additionally, for the same target, the set of sensors feeding the track changes in time, and therefore the

state vector incorporates new bias terms as it traverses the coverage of new sensors.

Track bias estimation is performed using ‘biased measures’, to estimate sensor biases from a predefined set of sensors, and ‘corrected measures’, from sensors whose bias terms were estimated and corrected in prior estimation stages (or which, according to their measurement model, have no bias). ‘Unusable measurements’ must not be used at this estimation stage, as the bias terms from their source sensors will be calculated in posterior stages.

As each measurement comes only from one sensor—only the bias terms from this sensor are projected into it—the Kalman filter may be arranged in a highly efficient manner, based on its reformulation as a set of coupled filters. The complete bias estimate may be arranged as a list of vectors (to be called ζ), the first ($\zeta[0]$) containing the position (X , Y) and velocity (V_x , V_y) in horizontal coordinates, and the others containing the bias estimate for each sensor, as represented in Fig. 2. With this structure in mind, when a measurement from a new sensor is processed, with its own bias terms, a new element ($\zeta[i]$) is appended to the ζ list. Each element of the estimate list could have a different size, as it will contain the bias terms potentially belonging to different kinds of sensors.

An additional key point is noting that the bias terms in each ζ element ($\zeta[i]$) will not always be the same elements of the vector ($\Delta\theta_s$, $\Delta\theta_t$, $\Delta\theta_{s-t}$), for the i th sensor, as sometimes, in order to solve observability problems, instead of estimating all bias terms, some linear combination of the bias terms will be calculated. Later steps of the bias estimation will allow the separation of those terms exploiting the diversity of target and sensor–target-related biases.

For the calculation of sensor- and target-related bias terms, not only the whole vector estimate from each target is necessary, but also the associated covariance matrix. Let us assume that we call $C_{i,j}$ to the cross-covariance between i th and j th estimation vectors ($\zeta[i]$ and $\zeta[j]$). It is clear, in this case, that $C_{j,i} = C_{i,j}^T$.

From this definition a list of matrices (to be called P) is built. The structure of P is depicted in Fig. 3. Two indexes are used to access the elements: $C_{i,j}$ is, in this structure, $P[i][j-i]$.

Track bias estimation is based on measurements from constant velocity movement segments. To start, the algorithm initialises the ζ and P lists. The position estimates (X and Y in Fig. 2) in $\zeta[0]$ are set to the horizontal positions of the first measurement in the first constant velocity segment. The velocity estimates (V_x and V_y in Fig. 2) are initialised using a rough estimate (based on monosensor measurements) of

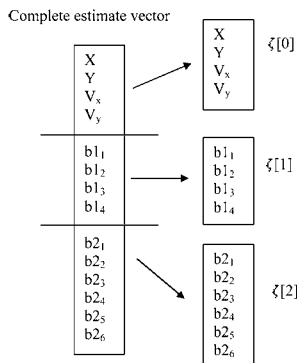


Fig. 2 List of estimates in a track bias estimator

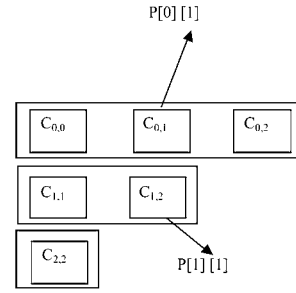


Fig. 3 Structure of P covariance list in local bias estimator

trajectory velocity. Then $P[0][0]$ is initialised to $C_{0,0}$, described in (3)

$$C_{0,0} = \begin{bmatrix} \sigma_{\text{pos}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{\text{pos}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\text{vel}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\text{vel}}^2 \end{bmatrix} \quad (3)$$

where σ_{pos} and σ_{vel} are constants to adapt the algorithm, set to high-enough values to provide freedom for any feasible bias term error projection in XY coordinates.

The algorithm processes in turn all the constant velocity segments in a given trajectory. After finishing the processing of a segment, the algorithm must reinitialise the $\zeta[0]$ vector and all the $P[0][i]$ matrices. In the case of $\zeta[0]$ and $P[0][0]$, it must be done as with the first measurement of the first constant velocity segment. The rest of the $P[0][i]$ matrices (cross-covariances between position/velocity vector and previously estimated sensor biases), must be reinitialised to zero matrices, as the new position estimate is assumed independent to previously calculated bias.

For each measurement in the constant velocity segment the algorithm checks if it belongs to a sensor not previously processed for bias estimation (with uncorrected bias terms) or not. In case it was not previously processed, the algorithm will

1. Add a new element, with zero values, to the ζ list, which will contain the bias terms related to this sensor.
2. Add a new element to all the P lists, a zero matrix with the adequate size (number of bias from the pair of sensors involved), at the final position.
3. Finally, create a new list ($P[N]$), which will be appended to the P list of lists, containing the initialisation of $P[N][0]$ ($C_{N,N}$). The contents of this matrix are sensor-type specific (let us call them C_{sensor}). Initially, they are diagonal matrices with very high values, in order to avoid having any impact on initialisation over the final bias estimation, as no prior knowledge over biases values is assumed.

Both in the cases that the processed measurement comes from a new sensor or not, the algorithm processes it with the following rearranged Kalman filter. Its processing steps are as follows:

1. Obtain the time since the immediately previous filtered measurement, to be called T .

2. Calculate position prediction matrix F , as

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

3. Predict estimates: all ζ elements remain constant (constant biases are assumed), but the one related with position

$$\zeta[0] = F\zeta[0] \quad (5)$$

4. Predict covariances: those P elements related with position need to be changed. Those are the covariances in the first list of P

$$\begin{aligned} P[0][0] &= FP[0][0]F^T \\ P[0][j] &= FP[0][j] \end{aligned} \quad (6)$$

5. If the algorithm is processing a ‘biased measure’, it must calculate the bias projection matrix H_b , dependent on both the sensor type and the measurement under analysis. H_b is directly related with $H_{\theta}(k)$ in (2); this relation can be derived from the relation between $\zeta[i]$ and vector $(\Delta\theta_s, \Delta\theta_t, \Delta\theta_{s-t})$ for each sensor.

6. Define a matrix to be called H_{pos} of the form

$$H_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (7)$$

7. Calculate the horizontal projection of the measured position, to be called x_m .

8. If the algorithm is processing a ‘biased measure’, calculate n as the index in the ζ list of the sensor providing the measurement.

9. For ‘biased measures’, calculate the residual of the Kalman filter as

$$r = x_m - H_{pos}\zeta[0] - H_b\zeta[n] \quad (8)$$

10. For ‘corrected measures’, calculate the residual of the Kalman filter as

$$r = x_m - H_{pos}\zeta[0] \quad (9)$$

11. Obtain the measurement covariance, projected in the horizontal plane, associated to the datum. This is a 2×2 matrix to be called R .

12. For ‘biased measures’, calculate the residual covariance matrix S as

$$\begin{aligned} S &= H_{pos}P[0][0]H_{pos}^T + H_{pos}P[0][n]H_b^T \\ &+ H_bP[0][n]^TH_{pos}^T + H_bP[n][0]H_b^T + R \end{aligned} \quad (10)$$

13. For ‘corrected measures’, calculate the residual matrix S as

$$S = H_{pos}P[0][0]H_{pos}^T + R \quad (11)$$

14. For ‘biased measures’, calculate two lists of N matrices. We will call them $B[i]$, with $0 \leq i < N$, and $BT[i]$. To

calculate them two different methods should be used, depending on the value of i

(a) If $i < n$

$$\begin{aligned} B[i] &= H_{pos}P[0][i] + H_bP[i][n-i]^T \\ BT[i] &= P[0][i]^TH_{pos}^T + P[i][n-i]^TH_b^T \end{aligned} \quad (12)$$

(b) If $i \geq n$

$$\begin{aligned} B[i] &= H_{pos}P[0][i] + H_bP[i][i-n] \\ BT[i] &= P[0][i]^TH_{pos}^T + P[i][i-n]^TH_b^T \end{aligned} \quad (13)$$

15. For ‘corrected measures’, calculate the same two lists of N matrices, but use the following equations

$$\begin{aligned} B[i] &= H_{pos}P[0][i] \\ BT[i] &= P[0][i]^TH_{pos}^T \end{aligned} \quad (14)$$

16. Calculate a list of Kalman gain matrices ($K[i]$) for the corresponding $\zeta[i]$ element, as

$$K[i] = BT[i]S^{-1} \quad (15)$$

17. Calculate the filtered estimate, through independent updates of the ζ elements, as

$$\zeta[i] = \zeta[i] + K[i]r \quad (16)$$

18. Finally, the filtered covariance structure must be updated. To update each element in the row a two-nested loops must be performed, with indexes $0 \leq i < N$ and $0 \leq j < N - i$. The corresponding P elements must be updated as

$$P[i][j] = P[i][j] - K[i]B[i+j] \quad (17)$$

After processing all the measurements from all the constant velocity segments of every multisensor track, the track bias estimates and their related covariance estimates for this estimation stage will be obtained.

3.2 Sensor bias estimation

Sensor bias estimation is the process of extracting the sensor-related biases ($\Delta\theta_s$) by averaging track bias estimators. To perform the averaging track bias filtered estimates $\zeta[i]$, referred to the same sensor, track bias estimates are reinterpreted as independent measures of the sensor bias, corrupted by a noise term (track bias estimation error) and by projections of target and sensor–target pair biases. So, the following measurement model is assumed, for each sensor (i), from a given track

$$\xi[i] \cong \Delta\theta_s + G_t[i]\Delta\theta_t + G_{s-t}[i]\Delta\theta_{s-t}[i] + \varepsilon_s \quad (18)$$

where:

- $\xi[i]$ is $\zeta[k]$, where k is the element in ζ list related i th sensor.

- $(G_i[i], G_{s-t}[i])$ are the bias projection matrices from the target and sensor–target pair bias terms, respectively to sensor bias terms coordinates, with i being the sensor index.
- ϵ_s is the track bias estimation error in sensor-related bias terms, which can be obtained from P matrix.

To perform the estimation of all sensor biases, a centralised Kalman filter processing every track bias estimate could be used. Quite often it is much more effective, in terms of computational load, to split it into several parallel filters, suited to a group of sensors. This artificial partition of the problem must take into account the potential observability problems, usually related to the sensor types. The partition also reduces the overall quality of the estimation, so there is a fundamental trade-off between quality and computational load to be analysed in each application.

Those Kalman filter states contain all the bias terms of the sensors of interest, assuming constant dynamic prediction model, and no plant noise. To perform the averaging, the interest sensor bias elements of ζ list must be extracted, to obtain the track bias ‘measurement’ vector for the sensor group (ξ), and the corresponding P terms must be arranged to obtain the related ‘measurement’ covariance matrix

$$\xi = [\xi[i_1]^T \quad \xi[i_2]^T \quad \dots \quad \xi[i_n]^T]^T$$

$$\Pi = \begin{bmatrix} P[i_1][0] & P[i_1][i_2 - i_1] & \dots & P[i_1][i_n - i_1] \\ P[i_1][i_2 - i_1]^T & P[i_2][0] & \dots & P[i_2][i_n - i_2] \\ \vdots & \vdots & \ddots & \vdots \\ P[i_1][i_n - i_1]^T & P[i_2][i_n - i_2]^T & \dots & P[i_n][0] \end{bmatrix} \quad (19)$$

In sensor bias estimation target biases and sensor–target pair biases are modelled as a noisy error component. Therefore from (18) and (19), the measurement covariance matrix R_{track} (associated to ξ) for Kalman filter will be: (see (20))

where:

- R_t is a covariance matrix modelling the lack of knowledge over target-related bias terms ($\Delta\theta$).
- R_{s-t} is a covariance matrix modelling the lack of knowledge over sensor–target pair related bias terms ($\Delta\theta_{s-t}$).

This measurement model, together with the postulated prediction model, may be used to derive the sensor bias Kalman filter, incorporating track bias estimates after the track processing is terminated.

3.3 Target and sensor–target bias estimation

Target-related biases should be estimated to complete bias correction. A reduced bias vector, containing only estimates of $(\Delta\theta, \Delta\theta_{s-t})$, has to be calculated at this step. Two approaches may be used in an offline system:

- Exploiting the information on the track bias estimate, removing previously calculated sensor-related biases and assuming that the remaining components are because of target-related biases, which could be averaged.
- Correct raw measurements with sensor-related biases and perform a new round of bias estimation.

Using the second approach, target bias estimation would be similar to track bias estimation, using reduced bias vectors and defining an associated list of vector ζ and of covariances P . Three cases may occur:

- Several measurements of a same type, only biased because of common target bias will be related to a common element of the new ζ list (which could be seen as a ‘virtual sensor’).
- If the measurements are only biased because of a sensor–target pair bias term, the situation is equivalent to that of track bias estimation, with each sensor related to a ζ list element.
- In case a sensor–target pair induces both target and sensor–target biases, (8), (10), (12) and (13), will be modified to take into account both potential sources of bias in measurements, one from the target bias ‘virtual sensor’ and another one from the actual sensor–target pair, related to the sensor.

In a real-time on-line tracker this estimation must be done in parallel with target tracking, to facilitate fast alignment of measurements and reduction of false manoeuvre detection.

Target and sensor–target oriented bias estimations are derived from a fewer measurements than sensor bias estimation, and so convergence is in general slower. The proposed bias estimation should assess its own quality, using target and sensor–target oriented bias estimator covariances, to set the target bias estimations to zero (default value) if they have not converged.

4 Methodology application and comparative evaluation

To clarify our bias architecture and compare its results with other approaches in literature, we will first apply it and evaluate the results in an example problem. This system is based on a distributed sensor network with one kind of 2D sensor (‘radar’ in the example), providing measures in polar

$$R_{\text{track}} = \Pi + \begin{bmatrix} G_t[i_1]R_tG_t^T[i_1] & G_t[i_1]R_tG_t^T[i_2] & \dots & G_t[i_1]R_tG_t^T[i_n] \\ G_t[i_2]R_tG_t^T[i_1] & G_t[i_2]R_tG_t^T[i_2] & \dots & G_t[i_2]R_tG_t^T[i_n] \\ \vdots & \vdots & \ddots & \vdots \\ G_t[i_n]R_tG_t^T[i_1] & G_t[i_n]R_tG_t^T[i_2] & \dots & G_t[i_n]R_tG_t^T[i_n] \end{bmatrix} + \begin{bmatrix} G_{s-t}[i_1]R_{s-t}G_{s-t}^T[i_1] & 0 & \dots & 0 \\ 0 & G_{s-t}[i_2]R_{s-t}G_{s-t}^T[i_2] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G_{s-t}[i_n]R_{s-t}G_{s-t}^T[i_n] \end{bmatrix} = \Pi + \Sigma \quad (20)$$

coordinates, with a model comprising the following terms

$$\begin{aligned} R_k &= R_{id}(k) + \Delta R + \Delta R_j + n_R(k) \\ \theta_k &= \theta_{id}(k) + \Delta \theta + n_\theta(k) \end{aligned} \quad (21)$$

where:

- $(R_{id}(k), \theta_{id}(k))$ is the ideal target position for the k th measurement, in local polar coordinates.
- ΔR is the sensor range bias, a sensor-related bias term.
- ΔR_j is the target-induced (transponder) range bias of j th target. It is modelled as a sample of a zero-mean uniform distribution with maximum error of 75 m.
- $\Delta \theta$ is the azimuth bias, a sensor-related bias term.
- $(n_R(k), n_\theta(k))$ are measurement noise errors, with standard deviations of 100 m and 0.1° , independent of range.

Arranged according to (2), the measurement model for this example would be

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \end{bmatrix} &\cong \begin{bmatrix} x_{id}(k) \\ y_{id}(k) \end{bmatrix} + \begin{bmatrix} \sin \theta_k & R_k \cos \theta_k \\ \cos \theta_k & -R_k \sin \theta_k \end{bmatrix} \begin{bmatrix} \sin \theta_k \\ \cos \theta_k \end{bmatrix} \\ &+ \begin{bmatrix} \Delta R \\ \Delta \theta \\ \Delta R_j \end{bmatrix} + \underbrace{\begin{bmatrix} \sin \theta_k & R_k \cos \theta_k \\ \cos \theta_k & -R_k \sin \theta_k \end{bmatrix}}_{G_s(k)} \begin{bmatrix} n_R(k) \\ n_\theta(k) \end{bmatrix} \end{aligned} \quad (22)$$

The architecture to derive these data is just composed of a single stage including:

- Track bias estimation: its Kalman filter has an H_b matrix equal to $H_s(k)$, as both range-related bias terms are additive. The estimated bias terms are $\hat{\zeta}[i] = [\Delta R + \Delta R_j \ \Delta \theta]^T$.
- Sensor bias estimation, mixing data from the different track estimators, adding $75^2/3$ to range-related variances and cross-covariances in the 11 matrix, to derive R_{track} matrix: this is the result of (18) and (20) applied to this problem, assuming a ± 75 m uniform distribution of target bias.
- Sensor bias correction.
- Target bias estimation through a five states Kalman filter, including 2D position, 2D velocity and transponder bias. It is equivalent to the one used as track bias estimator, using an H_b matrix equal to $H_t(k)$.

4.1 Example application simulation results

The example application will be evaluated in a simulated scenario with three radars (circles in the diagram) and ten aircraft following constant velocity trajectories are depicted in Fig. 4. They are observed periodically every 5 s during 1500 s.

The biases (range-azimuth) and positions (X, Y) of the three radars are summarised in Table 1, while target-oriented biases are summarised in Table 2.

Fig. 5 shows sensor bias convergence and target bias convergence. Sensor bias results show the convergence of the sensor bias estimation along time, assuming that every single sensor bias estimate is obtained on the track bias estimates obtained until that time. Target bias estimate convergence assumes that sensor bias is corrected with the last sensor bias estimate before the target bias is derived. Results from different radars are depicted: radar 1-continuous

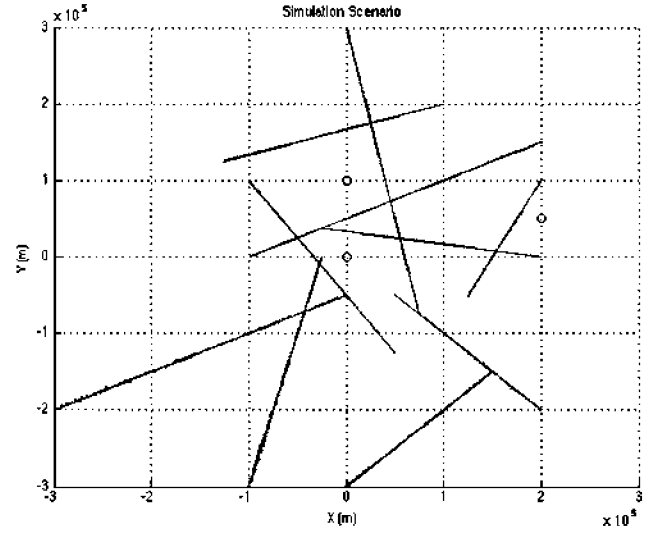


Fig. 4 Radar positions and target trajectories in simulated scenario

Table 1 Radar network parameters

Radar	Radar position (X, Y) in Km	Simulated biases (range in m, azimuth in degrees)
1	(0, 0)	(0, 0)
2	(200, 50)	(100, 0.1)
3	(0, 100)	(50, -0.05)

Table 2 Simulated transponder biases

Target	1	2	3	4	5	6	7	8	9	10
Bias, m	0	10	50	-20	-15	30	-50	-20	10	5

line, radar 2-dashed line, and radar 3-dashdot line. The central estimates in one run are depicted in black, while 2σ standard deviation bands around the central estimate are drawn in grey. These four figures show, respectively, range and azimuth sensor bias estimator convergence, target bias estimator for target number 7, and the error in Y coordinate position estimate for a Kalman filter processing sensor and target bias corrected measures. Even though sensor bias and target bias estimators contain residual bias error, they usually tend to compensate, leading to mostly unbiased tracking.

Convergence of sensor bias estimation is quite fast, and it does not improve much with time, owing to the presence of a transponder bias standard deviation imposing a minimum error level related to the number of targets in the scenario. Only by adding new targets to the estimation this component would be reduced. Owing to the small number of simulated targets, the lower limit for the standard deviation of the range bias estimation is quite high (of the order of 15 m for our simulation).

Regarding target bias estimation, its convergence is also fast, depending on the number of the integrated measures. In this scenario, a standard deviation around 5 m is obtained.

4.2 Comparison with classical approaches to bias estimation

Next we will compare our bias estimation and correction results with those obtained using estimation methods not

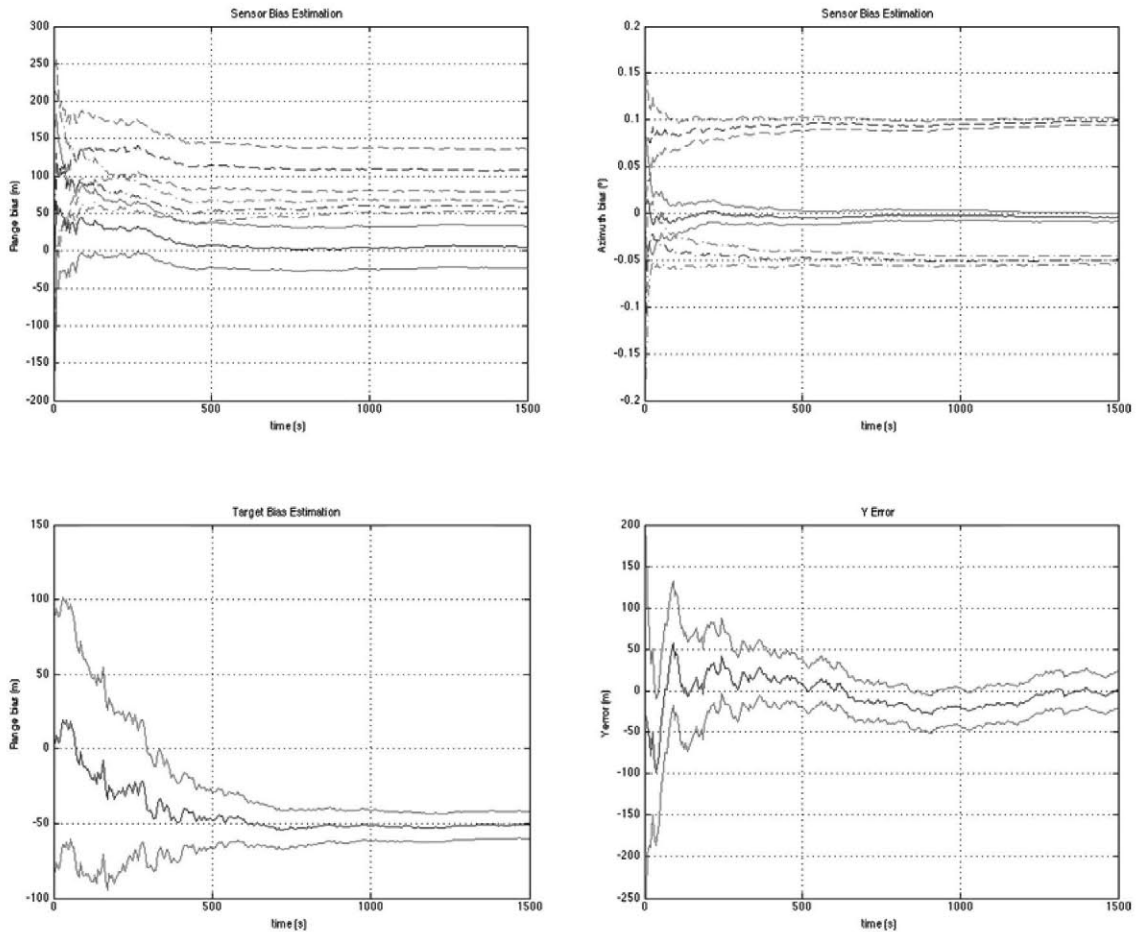


Fig. 5 Proposed system bias estimation convergence

taking into account the presence of target or sensor–target biases. The system to be used for the comparison is roughly equivalent to our proposal, using the same track bias estimator, followed by a sensor bias merging, which does not take into account the presence of target-induced bias and therefore does not add the $75^2/3$ term to range-related variances and cross-covariances in the \mathbf{I}_1 matrix to derive $\mathbf{R}_{\text{track}}$ matrix. Finally, this system does not estimate target biases. The results obtained for the same simulated scenario are in Fig. 6. The two graphics show the third range estimator convergence and the seventh target Y error for the same Kalman-tracking filter used in our system.

From these results we can see that, in general, this procedure is able to obtain quite similar estimators for the sensor bias estimation, although they may have higher residual biases and the covariances estimated by the sensor estimator are less consistent with the actual error distribution. Finally, as the target bias is neither estimated nor corrected, the measures remain biased, which in our simplified system without manoeuvres appears in the form of a biased track position, with inconsistent covariances. This is the most important result of the bias estimation and correction system. In a real system with a realistic tracking filter with manoeuvre detection, the presence of not

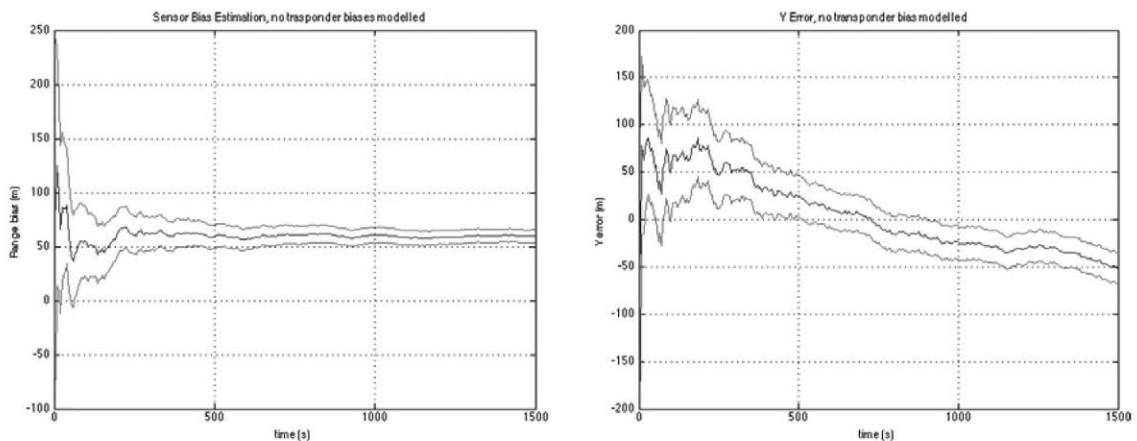


Fig. 6 Classical system bias estimation convergence

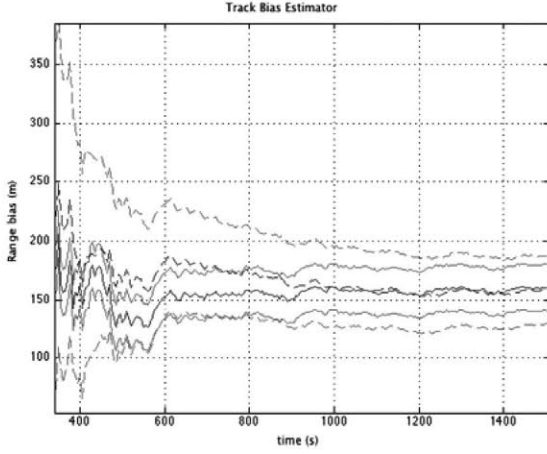


Fig. 7 Convergence of track bias basic algorithm

removed target bias would lead to and increased false manoeuvre detection rate.

Next, we will compare the performance of the basic track bias estimation procedure with that of a reference work in literature [5]. The model in the mentioned paper includes different terms than ours, so we implemented a version of the paper proposed algorithm restricted to our model (with only range and azimuth biases) for evaluation. Fig. 7 shows the results for our track bias estimation step method, compared to those obtained using the reference algorithm (dotted lines), for radar number 3, target number 3 and range estimate. The reference paper, as others using measure differences as the basis for their estimation, loses some information because of its need to split the measure chains to obtain independent pseudo-measures, which can be processed by a Kalman filter. The proposed method (and the proposal in [2]) does not suffer this problem and therefore it provides slightly better estimates. In any case, independently of the track bias estimation method used, the lack of decomposition of bias terms in sensor and target biases will not allow obtaining high-quality bias estimators.

5 Application to ATC reconstruction bias estimation

In this section, we will apply the described methodology to an offline ATC trajectory reconstruction system, describing the error models and resulting bias estimation and correction system. The proposed error models, and therefore the resulting algorithms, are those used in SASS-C v7/OTR. Modern ATC data fusion system relies on the following sensors data:

- Radar plots, from primary (PSR), secondary (SSR) and mode S radars, including enhanced surveillance [16, 17].
- Automatic dependent surveillance (ADS-B) data [18].
- Target reports from wide area multilateration (WAM) sensors [19].

5.1 ATC radar error models

There are mainly two types of ATC radars: primary (PSR) and secondary (conventional secondary, also known as SSR and mode S). They measure range and azimuth, and in the case of secondary radars, they also provide height from the aircraft barometric altimeter.

In the secondary radar error model, k th range-azimuth measurement (R_k, θ_k) include the terms in (23)

$$\begin{aligned} R_k &= (1 + K)R_{id}(k) + \Delta R + \Delta R_j + n_R(k) \\ \theta_k &= \theta_{id}(k) + \Delta\theta + \Delta\theta_1 \sin \theta_{id}(k) + \Delta\theta_2 \cos \theta_{id}(k) + n_\theta(k) \end{aligned} \quad (23)$$

where:

- $(R_{id}(k), \theta_{id}(k))$ are the ideal target position, expressed in local polar coordinates.
- ΔR refers to radar range bias, from drifts in the radar chain calibration [16].
- $(1 + K)$ is range gain, mainly dependent on propagation [17] and sensor clock drift effects.
- $\Delta\theta$ is azimuth bias, because of drifts in the rotation servomechanism [16].
- $(\Delta\theta_1, \Delta\theta_2)$ characterise the radar's azimuth eccentricity, related to maximum eccentricity and angle of maximum eccentricity. They model errors such as wind-related or mechanic fatigue-related asymmetries of rotation, or potential radar vertical apex drift with only an sinusoidal component (with amplitude and phase). Additional harmonics modelling these errors are not considered because of their lower amplitude and of the computational needed to estimate them [20, 21]. All previous bias terms are sensor related.
- ΔR_j refers to the transponder induced bias of j th aircraft, a target-related bias term. All the transponders are allowed to have a bias in the reply of up to ± 75 m for SSR and ± 37.5 m for mode S measurements [16].
- $(n_R(k), n_\theta(k))$ are measurement noise errors. Those errors come from different sources, such as thermal noise, quantification error, clock jitter or transponder reply jitter [16].

Primary radars share an equivalent model lacking ΔR_j term.

To translate range-azimuth-height measurement to the common stereographic plane where all bias calculations are performed, a non-linear coordinate transformation method [22, 23] is used. Although [11] proposes the use of ECEF coordinates for bias estimation, from our experience stereographic transformations and related linearisations can be successfully used for network coverage up to 5.000 km. This method implements a function to be called $f_{\text{Radar}}(\cdot)$. To project the error terms into the stereographic plane, a first order approximation of this transformation can be made, resulting

$$\begin{bmatrix} x_{k,0} \\ y_{k,0} \end{bmatrix} = f_{\text{Radar}} \left(\begin{bmatrix} R_k \\ \theta_k \end{bmatrix} \right) \quad (24)$$

There is also a potential timestamp bias, for old non-time synchronised radars, leading to an equivalent position bias aligned with velocity. Then, (X, Y) projected measurements will suffer an additional bias of the form

$$\begin{aligned} x_k &= x_{k,0} - V_X \Delta t \\ y_k &= y_{k,0} - V_Y \Delta t \end{aligned} \quad (25)$$

where:

- (V_X, V_Y) is the velocity vector of the target.

- Δt is the time bias for the radar, another sensor-related bias, coming from drifts in the radar clock.

From (23)–(25), it is possible to obtain a model coherent with (1) and (2)

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \end{bmatrix} &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \Delta \theta_s, \Delta \theta_t, \Delta \theta_{s-t}, n_s(k) \right) \\ &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \begin{bmatrix} \Delta R \\ K \\ \Delta \theta \\ \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta t \end{bmatrix}, [\Delta R_j], \square, \begin{bmatrix} n_R(k) \\ n_\theta(k) \end{bmatrix} \right) \\ &\cong \begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{H}_R(k) & \mathbf{H}_{R-t}(k) \end{bmatrix} \begin{bmatrix} \Delta \theta_s \\ \Delta \theta_t \end{bmatrix} + \mathbf{G}_s(k) n_s(k) \end{aligned} \quad (26)$$

For ATC radars, provided that a complete development of the Jacobians is performed, linearisation is good enough to allow accurate bias estimation.

5.2 ADS-B error models

ADS-B is based on the broadcasting of aircraft navigation information to ground through a data-link [18]. With precise navigation systems (differential GPS or GPS), ADS-B measurements suffer mainly from a time-stamping error, leading to a time bias different for each aircraft. ADS-B time stamping is usually not performed at the aircraft, but on the ADS-B reception station. The ADS-B on-board equipment and ADS-B reception clocks might not be synchronised, as some of the delays introduced by data buffering in the aircraft or ground station may not be calibrated [24].

Therefore the k th position measurement (x_k, y_k) , obtained using the stereographic projection of ADS-B latitude, longitude and height measurements, may be modelled following the notation in (1) and (2) as

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \end{bmatrix} &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \Delta \theta_s, \Delta \theta_t, \Delta \theta_{s-t}, n_s(k) \right) \\ &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \square, [\Delta t_j], \square, \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} \right) \\ &= \begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix} + \mathbf{H}_{\text{ADS}} [\Delta t_j] + \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} \\ \mathbf{H}_{\text{ADS}} &= \begin{bmatrix} -V_X \\ -V_Y \end{bmatrix} \end{aligned} \quad (27)$$

where

- (V_X, V_Y) is the velocity vector of the target.
- Δt_j is the time offset for j th aircraft. It is a target-related bias term.
- $n_x(k), n_y(k)$ are the measurement noise errors. Their covariance is dependent on the navigation system quality, which is coded in the ADS-B message using NAC_p (navigation accuracy code for position) and on the quantification step in ADS-B messages.

5.3 WAM error models

WAM performs time difference of arrival (TDOA) [19] estimation to calculate target position, based on the emission of random signals by the aircraft (secondary transponder squitters, responses to nearby radars etc.) and its reception by a ground-based station network. The error of multilateration is a complex function of the geometry of the receiving stations and transmitter, the timing accuracy of the receiving stations, the accuracy of the synchronisation process of the receiving sites, and of propagation effects. WAM has internal calibration means, as without them no position estimation would be possible. So we are dealing with remaining errors after this calibration. Modelling of WAM errors is a subject under active research, a description of the main error terms may be found in [25].

In this paper, we assume bias is constant in time, and after discretisation of the space in 3D cells, bias is assumed constant within each cell. Then, the k th position measurement (x_k, y_k) , obtained using the stereographic projection over latitude, longitude and height measurements, may be modelled following (1) and (2) as

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \end{bmatrix} &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \Delta \theta_s, \Delta \theta_t, \Delta \theta_{s-t}, n_s(k) \right) \\ &= f_{\text{sensor}} \left(\begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix}, \begin{bmatrix} \Delta X(1) \\ \Delta Y(1) \\ \vdots \\ \Delta X(N) \\ \Delta X(N) \\ \Delta t \end{bmatrix}, \square, \square, \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} \right) \\ &= \begin{bmatrix} x_{\text{id}}(k) \\ y_{\text{id}}(k) \end{bmatrix} + H_{\text{WAM}} \begin{bmatrix} \Delta X(1) \\ \Delta Y(1) \\ \vdots \\ \Delta X(N) \\ \Delta X(N) \\ \Delta t \end{bmatrix} + \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} \\ \mathbf{H}_{\text{WAM}} &= \begin{bmatrix} 0_{2 \times 2} & \cdots & I_{2 \times 2} & 0_{2 \times 2} & \begin{bmatrix} -V_X \\ -V_Y \end{bmatrix} \end{bmatrix} \end{aligned} \quad (28)$$

where

- (V_X, V_Y) is the velocity vector of the target.
- $(\Delta X(n), \Delta Y(n))$ is the X, Y bias for n th cell in the cell list. They are a set of sensor-dependent bias.
- Δt is the time bias, equal for all aircraft and cells. It is a sensor-dependent bias.
- $(n_x(k), n_y(k))$ are the noise components in stereographic plane. They are correlated, and their covariance may be obtained from the relative target to station network geometry analysis. It is usually provided together with the WAM measurement.
- $0_{2 \times 2}$ is a zero matrix of dimension 2×2 , and $I_{2 \times 2}$ is the identity matrix of the same dimension, which is positioned within H_{WAM} at the place corresponding to the n th cell where the measurement lies.

It should be noted that the cell list could be quite large. Therefore there are no measurements from many of those cells, being the bias unobservable.

5.4 Bias estimation and correction methodology for ATC reconstruction

Track–sensor–target bias estimation process can be done in several stages, with different measurements feeding the bias estimator, in order to increase integrity. For instance, in OTR, radar data were assumed to be more reliable, because of the longer operational experience. The resulting processing order is the one showed in Fig. 8. ADS-B and WAM bias estimation and their corrections are independently performed. Radar bias estimation and correction are a prerequisite for both ADS-B and WAM bias estimation, which is a limitation of current SASS-C v7/OTR to be addressed in future versions. In both cases, radar plots are assumed ‘corrected’.

5.5 Track bias estimation

For secondary radars, there is lack of observability of range bias and transponder delays terms separately. The radar track bias estimator will estimate their sum. For each secondary radar the track bias estimate ($\zeta[i]$) will have a state vector containing $[\Delta R + \Delta R_j \ K \ \Delta \theta \ \Delta \theta_1 \ \Delta \theta_2 \ \Delta t]$. For primary radars, the bias vector will be $[\Delta R \ K \ \Delta \theta \ \Delta \theta_1 \ \Delta \theta_2 \ \Delta t]$. So, H_b in radar track bias estimation is H_R , as defined in (26).

ADS-B Track bias estimation assumes radar measurements are unbiased. The ADS-B track bias estimate has the state vector: $[\Delta t_j]$. Accordingly, H_b for ADS-B track bias estimation is H_{ADS} as defined in (27).

WAM track bias estimation is special, as each measure contains only information of the terms related to the WAM error cell that the target is traversing. The bias estimation terms for WAM track bias estimation are the values of the biases at each cell. To deal with the number of cells, OTR will increase the length of the bias vector as the data fusion system receives measurements from new cells. Using this

approach, each time the target enters a new cell, it is as if we were using a new virtual sensor to measure it, and three new bias terms will form a new element of the ζ_i list. So a virtual subsensor for each WAM cell is defined, where the bias terms are $[\Delta X(n) \ \Delta Y(n) \ \Delta t]^T$. All cells bias estimations contain a time bias estimate, although time bias is actually unique for all cells.

From this, H_b for this track bias estimator will be

$$H_b = \begin{bmatrix} 1 & 0 & -V_x \\ 0 & 1 & -V_x \end{bmatrix} \quad (29)$$

5.6 Sensor bias estimation

Radar sensor bias estimation averages radar track bias estimators to obtain the sensor bias estimates $[\Delta R \ K \ \Delta \theta \ \Delta \theta_1 \ \Delta \theta_2 \ \Delta t]$ for all radars. A Kalman filter, very similar to the one described in Section 4, but with additional sensor bias terms is used. In this filter, all the sensor bias terms from all the available radar are stacked in the state vector. Following the description in Section 3.2, the R_{track} matrix should include a term to be added to every range-related variances and cross-covariances in the S matrix. This term is related with the lack of knowledge of the transponder delay. Therefore all SSR and Mode S sensors track bias estimates will have a correlated error term (different and independent for SSR and Mode S) affecting range biases. This term appears in the algorithm as an additional variance equal to $\sigma_y^2 = 75^2/3$ m for SSR, and $\sigma_y^2 = 37.5^2/3$ m for mode S, assuming uniform error distributions of transponder delay in the observed fleet. These terms are used to build the elsewhere zero Σ matrix described in (20), to be used for radar bias estimation.

Additionally, fixed field transponders are used as a means to enable calibration of radar networks. These transponders location is known with high accuracy, and therefore the actual range and azimuth from all the radars receiving replies from it can be calculated. Averaging out all the range and

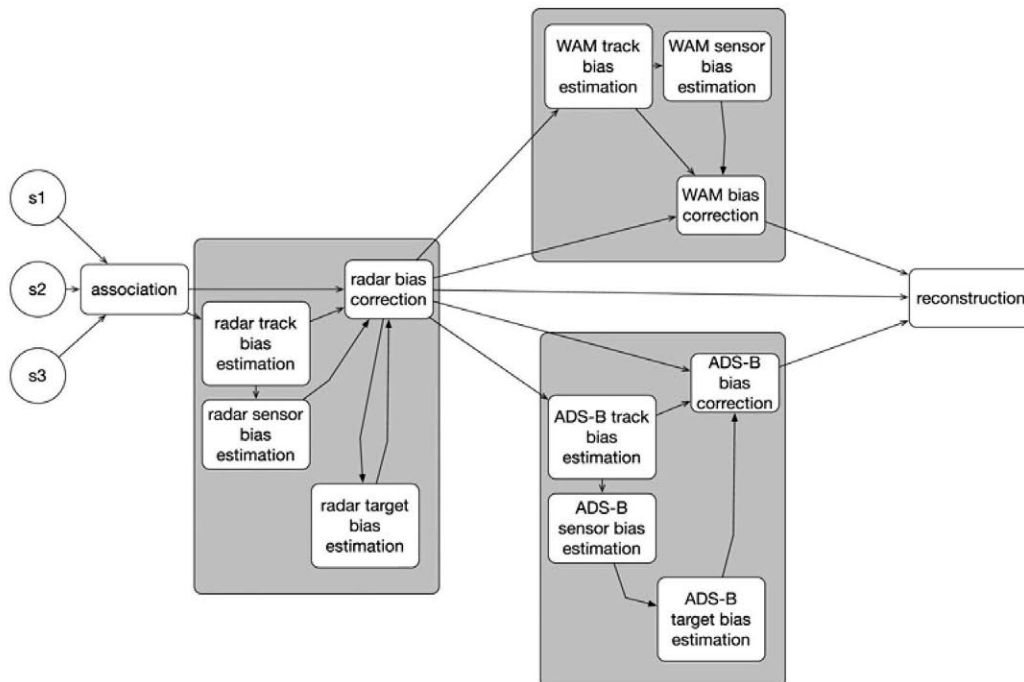


Fig. 8 Bias estimation and correction algorithms

azimuth measurement errors, a pair of observations combining several bias terms is available for each sensor, which can be incorporated to the radar bias estimation Kalman filter.

ADS-B sensor bias estimation is slightly different. If a given aircraft does not contain a constant velocity segment, ADS-B track bias estimator will not provide accurate ADS-B timestamp bias. The OTR algorithm calculates an average time bias for all ADS-B, and combines it with the available information from all the sensors taking into account its relative qualities, by using a new Kalman filter whose only state is time bias. Doing so with time bias means to assume that there is a mean time offset in the ADS-B fleet that can be averaged and used to enhance tracking quality of all sensors, and exploited for those ADS-B targets without any constant velocity segment.

Finally, WAM sensor bias estimation averages the WAM-related results from track bias estimators and obtains a unique time bias for the whole sensor by mixing all the time biases, and averaging track bias X - Y offsets per cell. It is implemented with a Kalman filter in which each track bias estimator from WAM is assumed to be a measurement of (i) the corresponding X - Y terms of the error cells traversed by the aircraft and (ii) the common to all cells time offset.

The approach is exactly the same as in the theoretical description; with the peculiarity that the Σ matrix is zero because of the lack of target bias terms.

The WAM sensor bias estimation state vector, according to the WAM bias model, contains $[\Delta X(1) \ \Delta Y(1) \ \dots \ \Delta X(N) \ \Delta Y(N) \ \Delta t]^T$.

In order to be able to estimate time bias, two aircraft or a single manoeuvring target, following two constant velocity segments, with different velocity vectors, must traverse at least one cell to produce the diversity needed to be able to distinguish XY offset from time bias.

5.7 Target bias estimation

After radar sensor bias estimation and correction, radar target bias estimation is performed with a filter equivalent to the one used for track bias, but with only two bias terms: transponder delays for Mode S and SSR modes. This filter does not have new terms as new sensors arrive, as transponder delay is independent of the interrogating sensor. The proposed algorithm has the restriction of needing constant velocity segments to derive this delay.

Regarding ADS-B, the implemented method uses the track bias as target bias if the track bias converged for this aircraft, and it will use the sensor bias if it did not converge (only correcting average time offset).

6 ATC reconstruction bias estimation results

SASS-C v7/OTR bias estimation has been tested with a complete set of simulation and real data scenarios. The estimation results are dependent on the actual position of simulated radars and the relative geometry of targets.

6.1 Simulated scenario results

First we will describe the results from a simulated scenario with 43 aircraft with a mixed fleet with SSR and mode S transponders, and a total amount of around 75 000 plots. SSR and Mode S radars have range and azimuth noise σ of 74 m and 0.09° , and PSR have a range and azimuth noise σ of 110 m and 0.18° .

Fig. 9 shows the radar coverage depicted in coloured lines, whereas the black lines depict simulated aircraft trajectories. There are two ADS-B stations, marked as AD-1 and AD-2 and a WAM sensor, marked as WM-1. In this deployment, there are a total of $10 \times 10 \times 3 = 300$ WAM bias cells, although there is traffic traversing only 45 of them.

In Table 3, there is a summary of range bias estimation results for the radars and their coordinates. The number is the identifier for the radar. In this simulation there are also ADS-B equipped aircraft (with assumed GPS or DGPS navigation systems) and WAM sensors (with a range error with a noise standard deviation in the order of 10 m, projected according to a geometric model based on a simulated WAM station deployment, using DOP concept [25]).

Table 4 summarises some transponder delay estimations and ADS-B estimations.

In general, results at this stage are not so good as in the first stage, as they accumulate errors from the previous correction, but it is clear that the bias estimation is converging to values close to the actual ones.

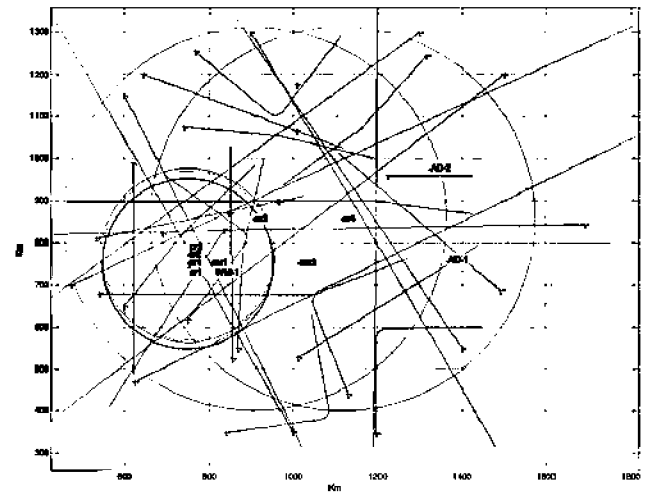


Fig. 9 Simulated scenario

Table 3 Simulated noise and radar bias estimation results

Radar number and type	Range bias, m	Estimated range bias, m	Estimated range bias 2σ band, m
PR1 (PSR)	50	36.07	± 22
MS2 (Mode S)	0	-1.29	± 28.9
SR2 (SSR)	0	-7.34	± 28.1
SR3 (SSR)	25	24.90	± 26.4
SR4 (SSR)	25	25.36	± 26.6
MS5 (Mode S)	150	150.44	± 26.6
SR5 (SSR)	150	169.24	± 38.1

Table 4 Simulated target bias estimation results

Transponder	Estimated transponder delay, m	ADS-B time bias, s	ADS-B estimated time bias, s
25	35.37	0	0.03
-25	-20.61	-0.5	-0.501
0	-2.03	0	-0.04

Regarding WAM, this simulation included zero WAM bias. WAM time bias was estimated as 0.011 seconds and, in the 45 cells where XY bias was calculated, the maximum bias error was of the order of 20 m, although most of them were smaller than 5 m, which also shows the convergence of this procedure.

6.2 Real data results

Fig. 10 shows the result of a pair of radar bias estimations from real data. It comprises real data sets for more than 10 mode S, secondary and primary radars, divided in 1 h exercises. The bias was estimated during 8 successive hours, in order to show bias estimation stability. The figure

shows range bias in metres, range gain, azimuth bias in degrees and time bias in seconds.

In these data there was a clear change in the estimation for hours number 3 and 4. After analysing the results, it became apparent that there had been a problem with the time stamping of several radars data (there was an offset exceeding 10 s for a certain subset of radars in a certain time interval), and therefore the constant bias assumption did not hold. Data regarding this scenario are provided to emphasise the need to have data consistent with estimation models.

A good evaluation of the consistency of the estimation is the distribution of the target bias. If the radar bias is not estimated in a consistent way, the transponder delay estimates will become large, and their distribution skewed. Fig. 11 depicts

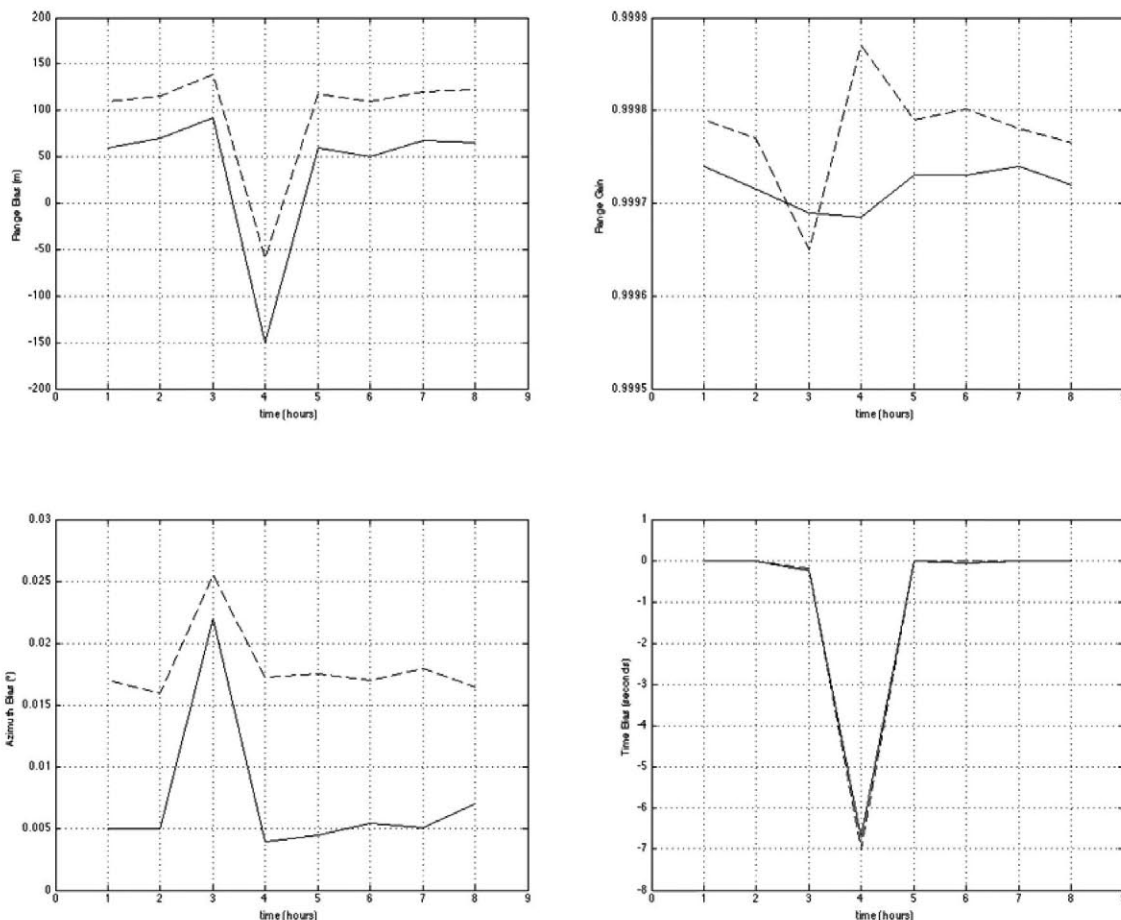


Fig. 10 Real data radar estimates from two radars in an 8 h-exercise

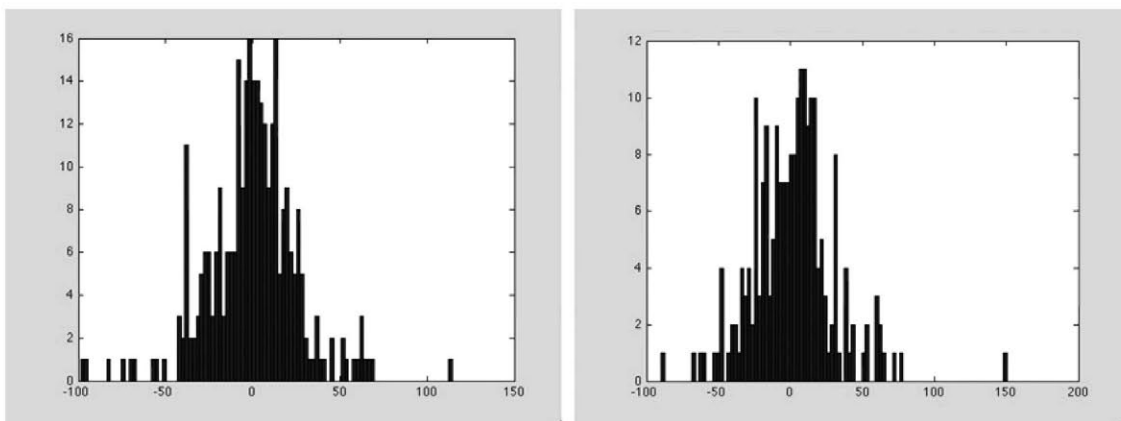


Fig. 11 Transponder delay estimates histograms for real traffic (SSR left, Mode S right)

a histogram of transponder delay estimates for a real scenario with nearly a thousand aircraft. Roughly one third of the transponder delays could not be estimated with good-enough quality, and therefore they are not included in the histogram. For bias correction they are left as zero values.

From the obtained results it is clear that the delay estimates show a distribution compatible with transponder 75 m maximum delay requirements, taking into account an error in the estimation of the order of 5–15 m for a typical trajectory. The estimates have approximately zero mean. Apart from this, mode S replies often suffer errors larger than the maximum for this kind of transponder (37.5 m).

7 Conclusions

This paper describes a generic architecture to estimate the bias of a set of sensors, and its application to an operational ATC reconstruction system. The architecture is based on a problem-specific multistage estimation procedure, in which biases for groups of sensors are calculated by performing four successive processes in each stage: (i) track bias estimation, (ii) sensor bias estimation, (iii) target and sensor–target pair bias estimation and (iv) bias correction. This approach extends the previous works to new types of sensors, enabling the reordering of bias estimation stages taking into account integrity assumptions or even on-line integrity assessments.

The results show that our approach is effective to calculate and correct bias, having a competitive performance when compared to other proposals in literature, and that it can be effectively applied to a complete complex application like SASS-C v7/OTR.

8 Acknowledgments

The Authors thank Dr. Emmanuel Voet from EUROCONTROL for his valuable comments and support during this research. This work has been funded under contract TRES by EUROCONTROL and under grants TEC2008-06732 and TEC2011-28626, and CONTEXTS S2009/TIC-1485 by the Spanish Government and the Regional Government of Madrid respectively.

9 References

- 1 Lin, X., Bar-Shalom, Y.: 'Design of an interacting multiple model algorithm for air traffic control tracking', *IEEE Trans. Control Syst. Technol.*, 1993, **1**, (3), pp. 186–194
- 2 Portas, J.A.B., Herrero, J.G., Vela, Gd.M.: 'New approach to online optimal estimation of multisensor biases', *IEE Proc. Radar Sonar Navig.*, 2004, **151**, (1), pp. 31–40
- 3 Blom, H.A.P., Hogendoorn, R.A., van Doorn, B.A.: 'Design of a multi-sensor tracking system for advanced air traffic control', in Bar-Shalom, Y. (Ed.): 'Multitarget-multi-sensor tracking: applications and advances, vol. 2' (Artech House, 1990, reprinted by YBS Publishing, 1998), pp. 31–64
- 4 Rafati, A., Moshiri, B., Rezaei, J.: 'A new algorithm for general asynchronous sensor bias estimation in multisensor-multitarget systems'. Proc. 10th Int. Conf. on Information Fusion, Quebec, Canada, July 2007
- 5 Lin, X., Bar-Shalom, Y., Kirubarajan, T.: 'Multisensor-multitarget bias estimation for general asynchronous sensors', *IEEE Trans. Aerosp. Electron. Syst.*, 2005, **41**, (3), pp. 899–921
- 6 Besada Portas, J., Garcia Herrero, J., de Miguel Vela, G.: 'Radar bias correction based on GPS measurements for ATC applications', *IEE Proc. Radar, Sonar and Navig.*, 2002, **149**, pp. 137–144
- 7 Pyung Soo, K.: 'Separate-bias estimation scheme with diversely behaved biases', *IEEE Trans. Aerosp. Electron. Syst.*, 2002, **38**, pp. 333–339
- 8 Ristic, B., Okello, N.: 'Sensor registration in ECEF coordinates using the MLR algorithm'. Proc. Sixth Int. Conf. on Information Fusion Queensland, Australia, July 2003, pp. 135–142
- 9 Nabaa, N., Bishop, R.H.: 'Solution to a multisensor tracking problem with sensor registration errors', *IEEE Trans. Aerosp. Electron. Syst.*, 1999, **35**, (1), pp. 354–363
- 10 Qi, Y., Jing, Z., Hu, S.: 'General solution for asynchronous sensor bias estimation'. 11th Int. Conf. on Information Fusion, July 2008
- 11 Okello, N.N., Challa, S.: 'Joint sensor registration and track-to-track fusion for distributed trackers', *IEEE Trans. Aerosp. Electron. Syst.*, 2004, **40**, (3), pp. 808–823
- 12 Xiaofeng, S., Li, C., Andong, S.: 'Asynchronous multi-sensor bias estimation with sensor location uncertainty'. Proc. 21st Annual Int. Conf. on Chinese Control and Decision Conf., Guilin, China, June 2009
- 13 Besada, J.A., de Miguel, G., Soto, A., Garcia, J., Alcazar, R., Voet, E.: 'TRES: multiradar-multisensor data processing assessment using opportunity traffic'. 2008 IEEE Radar Conf., Rome, 2008
- 14 Gelb, A.: 'Applied optimal estimation' (The M.I.T. Press, July 1982)
- 15 Kay, S.M.: 'Fundamentals of statistical signal processing: estimation theory' (Prentice-Hall, Inc., 1993)
- 16 Stevens, M.C.: 'Secondary surveillance radar' (Artech House, 1988)
- 17 Barton, D.K.: 'Modern radar system analysis' (Artech House, 1988)
- 18 Donohue, G.: 'A visionary look at aviation surveillance systems', *IEEE Aerosp. Electron. Syst. Mag.*, 1995, **10**, pp. 8–14
- 19 Galati, G., Leonardi, M., Tosti, M.: 'Multilateration (Local and Wide area) as a distributed sensor system: lower bounds of accuracy'. Proc. Fifth European Radar Conf., 2008
- 20 Hoogendoorn, R.A., Neven, W.H.L.: 'ARTAS: multisensor tracking in an ATC environment'. Sensor and Propagation Panel Symp., Lisbon, 1997
- 21 Fischer, W.L., Muehe, C.E., Cameron, A.G.: 'Registration errors in a netted air surveillance system'. Technical note 1980-40, MIT Lincoln Laboratory, 1980
- 22 Paradowski, L., Kowalski, Y.Z.: 'An effective coordinates conversion algorithm for radar-controlled anti-aircraft systems'. MIKON'98, 12th Int. Conf. on Microwaves and Radar, 1998, vol. 3, pp. 771–775
- 23 Torge, W.: 'Geodesy' (De Gruyter, 2001)
- 24 Minimum operational performance standards for 1090 MHz extended squitter automatic dependent surveillance – broadcast (ADS-B) and traffic information services – broadcast (TIS-B). RTCA document DO-260A, 2006
- 25 Neven, W.H.L., Quilter, T.J., Weedon, R., Hogendoorn, R.A.: 'Wide area multilateration'. Report on EATM TRS 131/04. Version 1.1, August 2005