Analysis of data fusion architectures and techniques in the development of an A-SMGCS Surveillance prototype[*]

José M. Molina
Jesús García
Dpto. Informática
Universidad Carlos III de Madrid.
Leganés, Madrid, SPAIN
molina@ia.uc3m.es
jgarcia@inf.uc3m.es

José R. Casar
GPSS-DSSR- ETSIT
Universidad Politécnica de Madrid
Madrid, SPAIN
jramon@grpss.ssr.upm.es

Abstract - *The work presented here addresses a key aspect of the data fusion process carried out by A-SMGCS Surveillance, needed to take maximum advantage from the simultaneous use of diverse detection and measurement technologies with complementary characteristics. The core function, Surveillance, must collect and fuse information from available sensors and information systems. The accuracy, coverage and refreshment rate in all-weather conditions must be high enough to satisfy the requirements. In a real application, this function will be constrained to deal with the specifications of output sensor data, formats, accuracies, etc. for the available sensors in the airport. In this work, we present a comparative analysis of data fusion architectures and some alternative algorithms to develop a real system deployed in Spanish airports, analyzing the capabilities and problems of different types of solutions*

Keywords: Fusion Architectures, ASMGCS, Airport Surveillance, Data Sensor Integration

1 Introduction

Modern Air Surveillance systems [1] are composed of several sensors (primary and/or secondary radars, passive sensors, acoustic sensors, image sensors, etc.) that provide data (detections, attributes, tracks) of every element in the covered environment.

Modern surveillance sensors have increasing capabilities to collect data, which are required to cope with ever increasing complex tactical environments (either military or civil). This places enormous additional burden on the system´s operator (fighter pilot or ATC controller). Multisensor integration is becoming an essential aspect of modern surveillance systems, which are being designed to use a network of multiple geographically dispersed sensors, applying sensor fusion concepts to improve their performance.

The fused data is presented to controllers after being merged at the fusion center. In the fusion center [2], the data fusion combines detections from sensors in an optimal way [2]. Multisensor integration is used to collect the information necessary to develop, by means of data fusion techniques, the perception of the scenario situation [3].

The application of this fusion techniques in Airport Surveillance functions are inside the ASMGCS concept. Advanced Surface Movement Guidance and Control Systems (A-SMGCS) [4][5][6] requires the surveillance of all aircraft and vehicles in the airport movement area. The system provides controllers (and potentially pilots) The Surveillance function provides a periodically updated synthetic image reflecting the current traffic state on the airport surface and close airspace, generating besides the output data to be used by the other functions of the A-SMGCS. with a display of the location of all surface traffic, enabling its separation and guidance in all types of weather conditions without reducing the number of operations or the level of safety. Therefore, A-SMGCS needs to encompass these functions:

- Surveillance: it must provide identification and precise position for aircraft, vehicles and objects (cooperative and non cooperative) within the movement area of the airport. This information has to be updated in real time, for the guidance and control functions purposes. Besides, surveillance must be immune to the weather conditions and the topography. This function is basic for the system, being a prerequisite for the others.

- Routing: it manages traffic routes on the surface, selecting “optimal” paths.
- Control: is on charge of preventing, monitoring, detecting and resolving all conflict types.
- Guidance: it indicates the pilots the routes they must follow, according to routing function results.

[*] This work has been funded by the Spanish CICYT contracts: TIC2002-04491-C01/02 and CAM (07T/0034/2003 1)

This paper describes the design of a fusion architecture using the deployed sensors in Spanish airports as the first step towards the implementation of an A-SMGCS prototype for Madrid/Barajas Airport. Nevertheless, one of the main requisites in this design is that it should be easily deployed in other airports (AENA currently manages all Spanish airports). Generally, proposed architectures are based on the centralization at the fusion center of all the measurement. In this work we analyze different architectures using a complete simulation of the airport layout with several scenarios.

In section 2 an analysis of real sensors deployed in Spanish airports is presented, in particular, the implications of adopting the ASTERIX format in the available data to process. Section 3 describes the fusion architectures designed. Section 4 presents some representative results of system behavior using several scenarios. Finally, paper conclusions are presented in section 5.

2 Airport Deployed Sensors

The targets moving on the airport are general and commercial aviation aircraft, and surface mobiles, such as fuel trucks, luggage convoys, buses and cars. The targets more important for us are the aircraft, but tracking other targets is also important, as far as they can compromise aircraft safety. In fact, A-SMGCS is in charge of increasing the safety of aircraft, by monitoring all kinds of traffic and providing directives to control aircraft on ground. To do that, it needs kinematic information of all aircraft and of those surface mobiles traversing airport areas in which they can compromise aircraft safety. Additionally, obtaining identification information of aircraft is necessary to be able to provide the control directives to the correct aircraft.

One of the key elements in any surveillance function is the type of sensors used to obtain measurements of the targets. Two main kinds of sensors can be used in this context:

- Cooperative sensors: they are based on the existence of on-board equipment, which helps the sensor in its task of detecting targets, providing identification, and measuring its position.
- Non cooperative sensors: they do not demand any help from the target. Usually, they do not provide identification information.

In next table we will show a list of the most usual sensors being used for airport surveillance, to compare their main features. The sensors described are Surface Movement Radar (SMR) [7], Multilateration systems (MS)[8][9], differential GPS broadcasted through a digital data-link (DGPS), ADS-B and, finally, TV. Clear meteorological conditions means not too dense fog, rain or snow. In the table we provide for each sensor if it is cooperative or not, its ability to provide identification, which mobiles may be tracked with this sensor, and under which meteorological conditions the system is usable.

Table 1: Sensor characteristics.

Sensor	Cooperative	Id.	Mobiles	Meteorological
SMR	*No*	*no*	*all*	*all*
DGPS	*Yes*	*yes*	*equipped*	*all*
MS	*Yes*	*yes*	*equipped*	*all*
ADS-B	*Yes*	*yes*	*equipped*	*All*
TV	*No*	*no*	*all*	*clear*

ASDE Radar:

Data Source Identificator
Message type: Plot SMR.
Time.
Position in polar coordinates related to radar position
Position in Cartesian coordinates (filtering process result).
Track Estimated Velocity
Track Estimated Acceleration
Track Number
Track State
Orientation of Vehicle Head
Radius of circumference that includes the track (centered in track centroid)

S Mode:

Data Source Identificator.
Message type: Multilateration Measure
Time
Position in polar coordinates related to reference point
Position in Cartesian coordinates (filtering process result).
Track Estimated Velocity
Track Estimated Acceleration
Track Number
Track State
A-Mode (from A-Mode Aircrafts)
Aircraft Identification from S-Mode aircrafts
Heigh Measure (if available).

ADS-B:

Data Source Identificator.
Message type: ADS-B Measure
Time
Position in spherical coordinates
Position in Cartesian coordinates (filtering process result).
Track Estimated Velocity
Track Estimated Acceleration
Track Number
Track State
Aircraft Identification

Fig. 1. Data attributes in ASTERIX format.

In this work the types of sensors considered are: ASDE radar, ASR radar, S-Mode multilateration systems and ADS-B. These sensors are deployed in Spanish airports and, then, they ought to follow the ASTERIX format. In figure 1, the data attributes are summarized.

ASTERIX format includes the corresponding track for each plot, this means that each deployed sensor processes previously the plots and maintains its local tracks. The local processing enhances the information received in the fusion center, for example, the identification of the corresponding track is useful for primary data such as plots from ASDE radar. But, on the other hand, some information could be lost, for example, the plots non-associated to any local track are not received in the fusion center.

Using ASTERIX information the architecture of fusion center could be based on the combination of local tracks. In this case, the developed architecture should be a distributed one without the capability to manage the local processing of local tracks. For example: (1) the continuity errors in local tracks obtained from the Surface Movement Radar should be translated automatically to central tracks, or, (2) the local filter defines the quality of central tracks.

Considering the limitation of a "pure" distributed architecture, in this work, we propose an architecture that processed the low-level data received from sensors and, besides, improves the association process using the attributes of ASTERIX format (basically, the information about local tracks allows a code-based quick association, with the risk of error propagation from sensor processor to the fusion node). This architecture could be developed following a "pure" centralized philosophy, or not, although all the processing steps will be physically located in the fusion center.

In this way, we have developed an scheme that allows us to design different fusion architectures (centralized, distributed and hybrid) using the data received in ASTERIX format. The main goal of the scheme is to manage the data received in ASTERIX format (Data Processed in Figure 2) joint to sensor specifications (Sensor Data in Figure 2) in order to improve the local information received. Figure 2 shows this scheme.

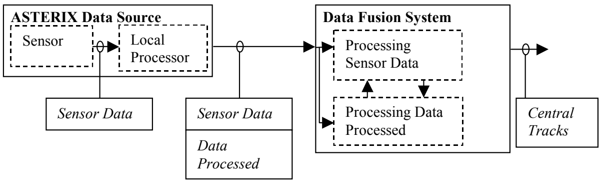


Fig. 2. General Scheme to use ASTERIX format

The proposed scheme works with the plots received using the processed data to facilitate the plot-central track assignation problem. If assignation or identification errors in local tracks are detected, the fusion system associates the plots directly improving the local information. For example, if data from the Surface Movement Radar appears with contradictory associations such as jumps in the track identification codes, the fusion system should associate again the plots to central tracks to avoid the translation of local errors to central tracks.

Fusion architectures are developed using this general scheme. So, when we describe several architectures, we describe the way to organize the information in the fusion center. Then, centralized architecture maintains only central tracks using ASTERIX data, distributed architecture maintains tracks for each sensor that are combined in central tracks, and hybrid architecture maintain tracks for each sensor like distributed architecture (for association purposes) and central tracks like centralized one. All the tracks in any architecture are generated and maintained in the fusion center, real local tracks are considered only as collateral information. Other collateral information are sensor models, airport map, dynamical models of surface targets, etc.

3 Fusion Architectures

Three different architectures could be developed: centralized, distributes, hybrid.

3.1 Centralized Architecture

The centralized architecture maintains a set of central tracks, $\{T_i\}$. The measures received from ASTERIX sensors, $\{P^{Sk}_j\}$, are associated to the central tracks and, then, filtered to actualize the track. In this architecture, the processes (association and filtering) works directly with measures (plots) and the additional information of the ASTERIX format is used in the association process to reduce the computational load. In Figure 3, the main processes of the centralized architecture are shown.

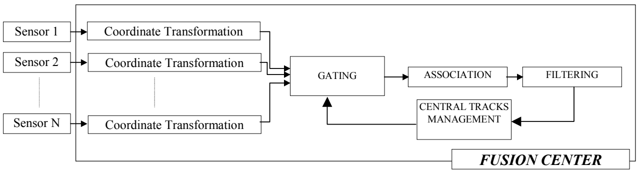

Sensor 1
Sensor 2
Sensor N
Coordinate Transformation
Coordinate Transformation
Coordinate Transformation
GATING
ASSOCIATION
FILTERING
CENTRAL TRACKS MANAGEMENT
FUSION CENTER

Fig. 3. Centralized architecture

The first step is the coordinate transformation function. In this step all ASTERIX format measures, $\{P^{Sk}_j\}$, (with coordinate values respect to sensor position, $\{S_k\}$), are transformed to unify the coordinates values with respect to the same global position, C, $\{P^C_j\}$. In a second step, temporal and kinematic compatibility is calculated for each pair measure-central track, $\{P^C_j, T_i\}$. Gating function evaluates the possibility of plot-to-central track association. The third step is the association function, where, a set of bidimensional matrixs (one for each sensor) are defined. Matrix rows are defined by tracks and columns are defined by sensor measures. Each matrix position contains: (a) if the association $\{P^C_j, T_i\}$ is possible, the value of the distance between measure and central track, or, (b) empty, if it is impossible. Munkres algorithm calculates association measure-central track that minimize the total distance. ASTERIX format information allows to reduce the matrix size and the computational load. Then, central tracks are actualized with the measures associated in the fourth step, the filtering function. The management of central tracks (generating new tracks, deleted track without measures o fused similar new tracks) is the final step.

The advantages of centralized architecture are:

- Optimize the position estimation for any sensor measure variance.
- Minimize the effects of a delay between the time when a maneuver begins and when it is detected, because maximize the refresh rate

The general problems (bandwidth occupation and computational load) of centralized architecture are not applied in our proposed scheme, because all the architectures are developed in the fusion center receiving and processing the same data. The major disadvantage of centralized architecture is devoted with systemic errors in sensor, due to the vulnerability to these errors and the difficulty for estimating them.

3.2 Distributed Architecture

Distributed architecture maintains central and local tracks. In this architecture, local tracks are maintained for each sensor and transformation, gating, association and filtering function are carried out locally to the sensor. The central tracks are the result of a fusion process over the local tracks that represent the same central track. This function is similar to the one carried out in the management of central tracks in centralized architecture. In this case the fusion process works at local track level, instead of measure level in the centralized architecture. Figure 4 shows the functions of the distributed architecture.

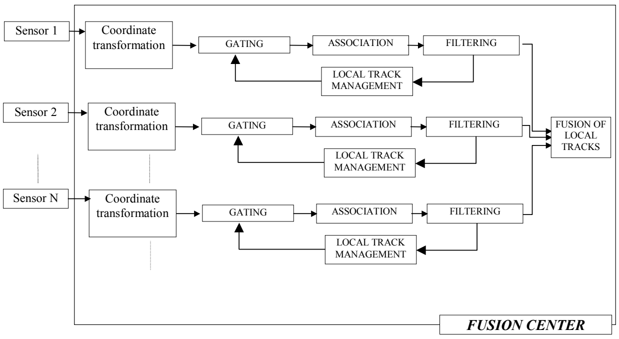


Fig. 4. Distributed architecture

The main advantage of distributed architecture is related with the distributed processing of measures that allow to adapt the functions specifically to each type of sensor. In this way, we can group the measures of surface radar instead of associate each individual measure as in centralized architecture. In our proposal, there are not advantages for the computational load in the fusion center as shown in the general scheme presented in section 2, although parallel execution of functions associated to each sensor is possible in a natural way. Another advantages are the capacity to calibrate sensor and to estimate bias easier than in centralized architecture. The main disadvantage is the loss of precision in the fused estimators.

3.3 Hybrid Architecture

The hybrid architecture is a combination of the previous ones. The fusion system combines the capacity to fuse at measure level or track level. In our implementation, local tracks are maintained to carry out the association of measures (as in the distributed architecture). But, central tracks actualization uses the measures instead of local tracks (as in centralized architecture). Figure 5 shows the functions of this architecture. The benefits of this architecture combines the ones of previous architectures. As advantages, the architecture presents: high accuracy, systemic error robustness, and the possible distribution of computational load, and, as disadvantages, a higher computational load because the measures are processed two times.

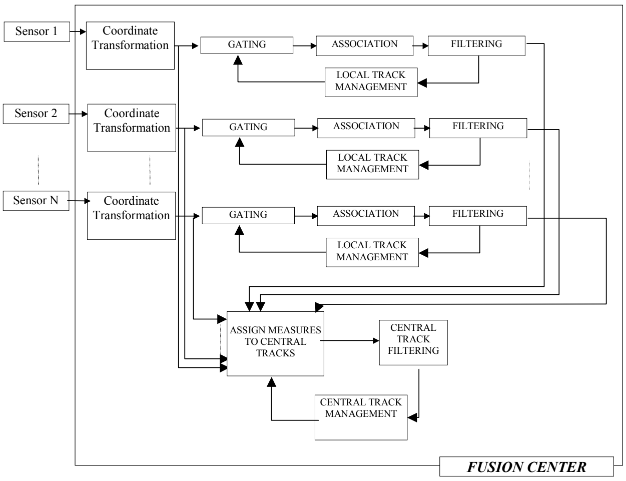


Fig. 5. Hybrid architecture

3.4 Shared Functions among architectures

The majority of functions considered in centralized, distributed and hybrid architectures are shared: coordinate transformation, gating, association, filtering and track management. Basically the functions are the same but in each type of architecture they work with local or central tracks. Figure 6 shows an scheme of the subfunctions included in these function.

Finally not-shared functions are: the "track fusion" function in distributed architecture, that is equivalent to the subfunction that appears in the track management function, and, the "assign measures to central track" function in hybrid architecture, that is an information compilation of local information providing for other functions and not a real function.

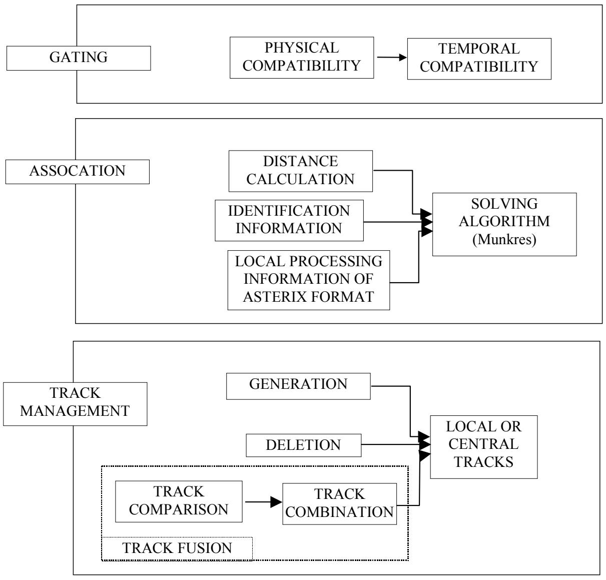


Fig. 6. Shared functions

4 Analysis of Fusion Architectures

In order to compare the three architectures, a simulation tool has been developed, without considering biases, systemic errors and some specific processing in sensors such as ASDE data extraction. The representative scenarios defined to test the architectures use an airport layout composed by a take-off/landing runway, an access taxiway and three exitways. Four sensors has been simulated: two surface radars (ASDE1 and ASDE2), a multilateration sensor S-Mode and an Approach Secondary Radar (ASR). In figure 7 the airport layout and the four sensors are represented, and in table 2 the sensors configuration are included.

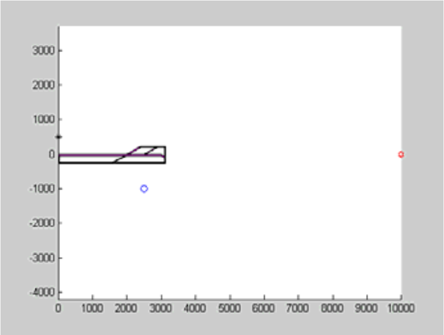


Fig. 7. Airport layout and deployed sensors

Table 2: Sensor Configuration

Sensor	**Position**	**Time Scan**	**Error**
ASDE1	(0, 500) m	1 s	σ_r=5m, σ_θ=0.05º
ASDE2	(2500, -1000) m	1 s	σ_r=5m, σ_θ=0.05º
Multi S Mode	4 reference locations (0,0)	1 s	4 references. Equivalent error: σ_x=5m, σ_y=5m
ASR	(10000, 0) m	4.8 s	σ_r=10m, σ_θ=0.09º

The evaluation of different scenarios over this experimental setup quantifies the architecture behavior in terms of:

- **Accuracy**: position and velocity mean square errors calculated for several situation: constant velocity sections, transversal acceleration sections (turning), section changes, stop and start movement and longitudinal acceleration in take off and landing. The transversal acceleration in every simulated maneuver take a value under 4 and 5 m/s^2, longitudinal acceleration to land under 2.0 m/s^2 and longitudinal acceleration to take off under 3.4 m/s^2. So the fusion system is asynchronous, the position and velocity predicted estimators have been analyzed at presentation times each 1 s. The additional error due to prediction is evaluated as an average for several simulations where the relations between presentation time and sensors time is changed. In resume, we evaluate:
    - Transversal error
    - Longitudinal error
    - Head error
    - Modulus error
- **Continuity**: this variable evaluates the robustness of the generated track in complex scenarios (near targets, high density of false alarms, etc.). We evaluate:
    - Commutation rate between tracks
    - Non-used data
    - Number of Generated Tracks
    - Number of Deleted Tracks
    - Number of Fused Tracks, when several tracks represent erroneously the same track
    - Number of Separated Tracks when one tracks represent erroneously several tracks
- **Computational Load:** the system calculates the computational time for each architecture in every scenario.

A database of scenarios has been created composed by several examples of aircraft movement in airport. These scenarios contains several possible configurations of false alarm rate. Besides the architecture could be defined with (1) different filters (conventional or including the airport layout information [10]), (2) different association logic (a conventional Munkres algorithm or using the identification included in ASTERIX format), and, (3) using all deployed sensors or a subset.

From this database in this paper we show the results for scenario 3 and scenario 8. The scenario 3 contains an aircraft landing and two vehicles braking to stop in two accessing taxiways. The first section of aircraft`s trajectory is a longitudinal acceleration in order to roll-off. Then, it makes a 45º turn to exit runway, and than a second 45º turn to follow other taxiway. The two vehicles have a stop and wait sections. Figure 8 shows the trajectories in the airport plane, figure 9 shows the trajectories in coordinate x and y with the time, and, figure 10 shows the aircraft spatial separation (by pairs) with the time, where it can be observed the short minimum distances between each vehicle and the aircraft.

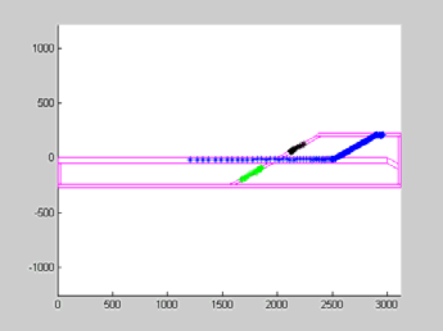


Fig. 8. Trajectories in scenario 3.

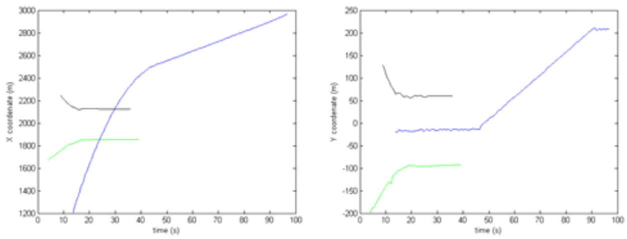


Fig. 9. Aircraft X and Y coordinates in scenario 3.

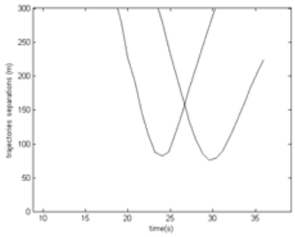


Fig. 10. Aircraft spatial separation.

Scenario 8 is defined with same aircraft than scenario 3 but with a false alarm rate in the zone with the minimum distance among the three aircraft. The false alarm rate in this zone is 10^{-3}. Figure 11 shows this zone.

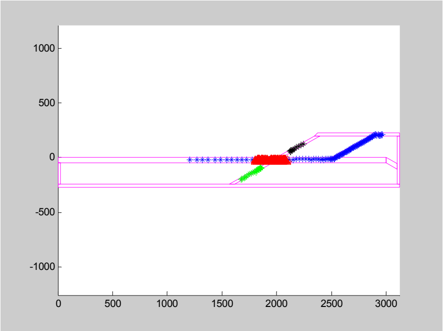


Fig. 11. Region with high false alarm rate in scenario 8.

Figure 12 shows the accuracy results for scenario 3 and figure 13 the continuity results. In figure 14 and 15 the same variables are shown for scenario 8. In the comparison, the centralized architecture is printed with solid line, the distributed one dashed line and the hybrid one dotted line. Blue color is used for map-guided tracking filter while red color for conventional IMM filter.

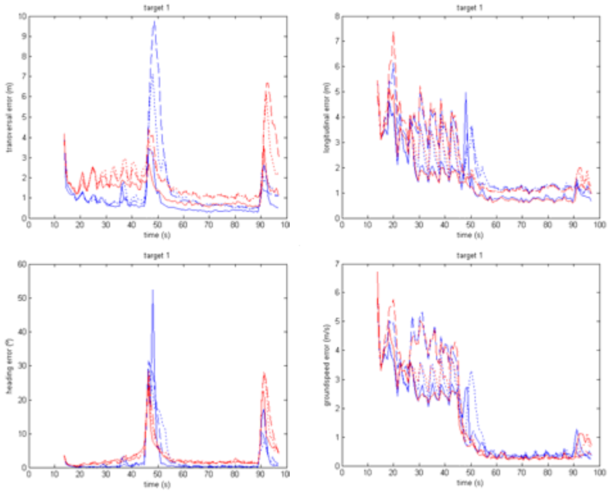
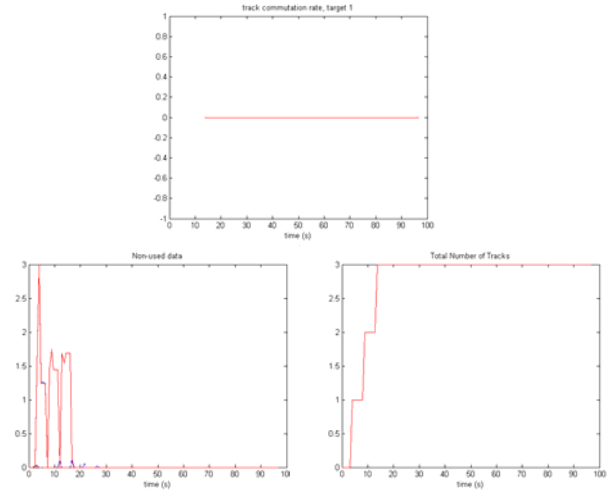


Fig. 12. Accuracy results for scenario 3.



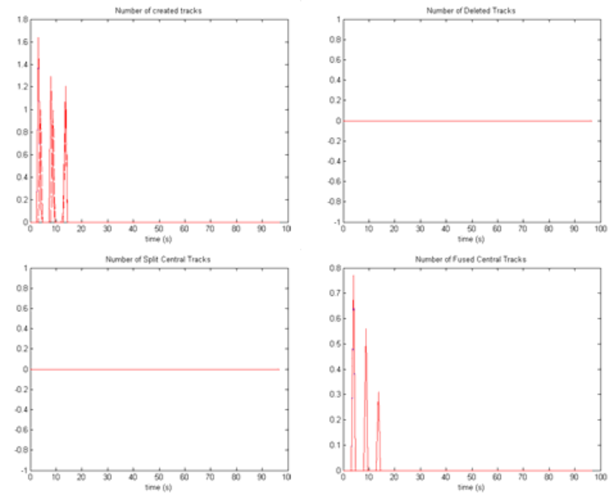


Fig. 13. Continuity results for scenario 3.

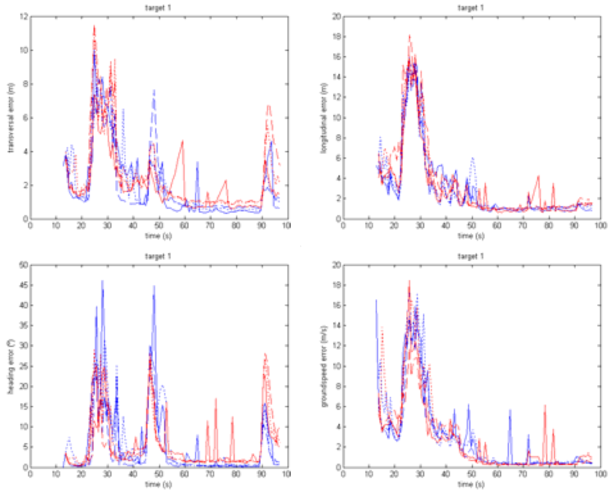
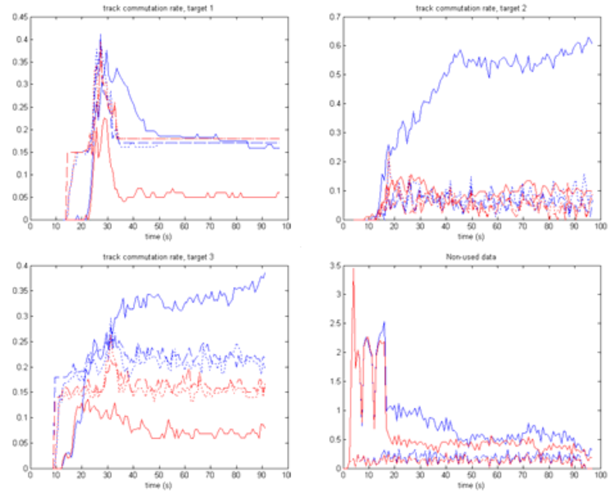


Fig. 14. Accuracy results for scenario 8.



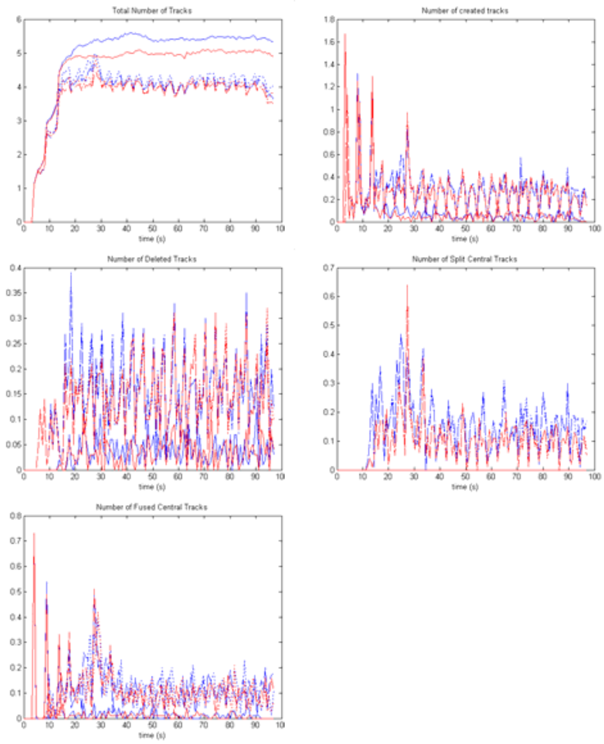


Fig. 15. Continuity results for scenario 8.

The computational load for the scenarios presented are shown in figure 16. The centralized architecture is again with solid line, the distributed one dashed line and the hybrid one dotted line. Black color is used for conventional association with NN-Munkres algorithm [1] while green is for direct association with ASTERIX code.

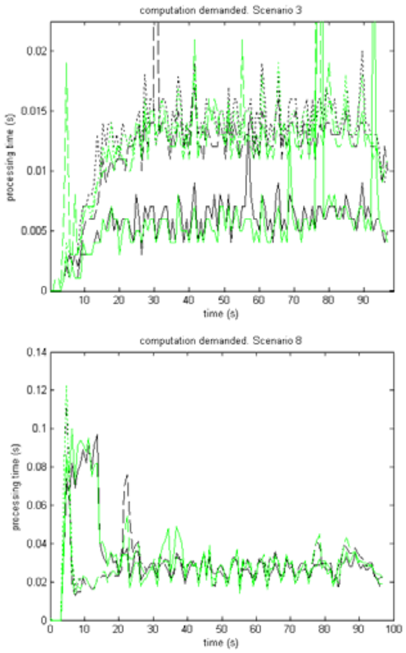


Fig. 16. Computational load for scenarios 3 and 8.

From the RMS error values, comparing the three architectures and filters, the centralized has the best performance for any type of filter. The distributed architecture degrades the performance due to higher extrapolation times, higher delays during maneuvering detections, and longer update rates producing noisier estimators, due to the interaction IMM logic (the transition probabilities among modes increase with time). Regarding the hybrid architecture, it has a higher error due to the fact that some out-of-sequence measures are lost, aspect not covered in this preliminary implementation.

With respect to the use of map in the filter, a significant improvement appears in the transversal and heading errors, due to the correction made with road orientations.

The figures of continuity performance indicate no differences in the scenario without false alarms. The performance is satisfactory under normal conditions, with maneuvers and separations adjusted to the minimum achievable by real aircraft. When the situation is forced to unreal separations, the centralized architecture showed a higher robustness. However, the behavior is different in areas with massive presence of clutter as in the scenario 8. An effect observed is that false tracks are more stable, with higher interactions with real tracks. The situation is worse with the map-guided filters, where false tracks are corrected and aligned with the airport map, increasing the chances to disturb tracks representing the real objects. This indicates that filtering false tracks should be addressed with a specific logic in the final design to reduce these problems.

The use of ASTERIX identifications to accelerate association is good with realistic scenarios, with moderate degradations appearing only under critical conditions. The effect on computational load is light, since the association is not significant compared with the effect of replicating the processing for all sensors (initialization, association, filtering and management). With false-alarms scenarios the higher burden is due to track management logic, occluding the differences among architectures and association options

5. Conclusions

Three fusion architectures have been analysed (centralized, distributed and hybrid) to be used with the available deployed sensors in Spanish airports as the first step towards the implementation of an A-SMGCS prototype for Madrid/Barajas Airport. The deployed sensors follow the ASTERIX normative, that defines a specific data format and implies the use of local processors in each sensor to maintain local tracks. The general scheme, defined in this paper, centralizes all the information about measures and sensors in the fusion center. Proposed architectures are based on this centralization that allows three different levels for the fusion process: measure level (centralized), track level (distributed) and measure/track level (hybrid).

The analysis of results shows the advantages and disadvantages of each architecture and the limitation/capacities in real environments, considerations to be taken into account in the design and development of the final solution to be implemented.

References

[1] Blackman, S.S., R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Dedham, MA, 1999.

[2] Waltz, E., J. Llinas, Multisensor Data Fusion, Artech House Inc., Norwood, MA, 1990.

[3] Bar-Shalom, Y., Multitarget Multisensor Tracking. Vols. I-II, Artech House Inc., Norwood, MA, 1992.

[4] European Manual of Advanced Surface Movement and Control Systems (ASMGCS). Draft. Volume 1: Operational Requirements. Draft Version 04. 08/24/2000.

[5] Proceedings of the ECAC APATSI and EC Workshop on A-SMGCS. Frankfurt, Germany. April 1994.

[6] Manual of SMGCS. ICAO". Doc. 9476-AN/927

[7] Schwabn C. E, Rost D. P.. "Airport Surface Detection Equipment" Proceedings of the IEEE, No. 2, February 1985.

[8] Rapport d'evaluation du système SYLETRACK. ADP. March 1997.

[9] Cooperative Area Precision Tracking System (CAPTS). Final report of test results. Frankfurt Airport. Airsys ATM Gmbh. January 1998.

[10] García, Jesús, Juan A. Besada, José R. Casar. Use of Map Information for Tracking Targets on Airport Surface. IEEE Transactions on Aerospace and Electronic Systems, Vol. 39, Nº 2. April 2003. pp 675-694.