Design of an A-SMGCS prototype at Barajas airport: available information and architecture*

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

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

Abstract - This paper describes the design of a fusion architecture using the sensors deployed on Madrid/Barajas airport as the first step towards the implementation of an A-SMGCS prototype (Advanced Surface Movement Guidance and Control System), providing Surveillance and Monitoring capabilities. The airport targets are general and commercial aviation aircraft and surface mobiles, such as fuel trucks, luggage convoys, buses and cars. Aircraft are the most important targets, but tracking other targets may be also important, as far as they can compromise aircraft safety. In fact, the goal of A-SMGCS is increasing the safety of aircraft, by monitoring all kinds of traffic and providing directives to control aircraft on ground. Additionally, aircraft identification is necessary to be able to provide control directives

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

1 Introduction

The application of fusion techniques [1][2][3] 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.

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].

In a previous work, presented at FUSION 2004, three alternative fusion architectures were analysed (centralized, distributed and hybrid) for this type of applications. From this previous work, we have developed a centralized architecture for data fusion, adapted to the airport configuration. This configuration includes the current taxiways description, the actual routing strategy for commercial flights and the location and characteristics of the available sensors.

In this work the types of sensors considered are: ASDE radar, ASR radar, Mode-S multilateration systems, and short-range radars (Millimetre Wave Sensors). These sensors will be deployed in Madrid-Barajas airport and, then, they will be integrated following the ASTERIX format for communications [13]. The paper will describe the global architecture of designed prototype, including the description and representation of airport information: layout, ground routes, sensor locations and configurations. The functional description of the implemented data fusion and monitoring systems will be included. The monitoring function will make use of the data fusion output together with the airport representation, available routes, and operational rules.

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). The A-SMGCS prototype follows the general scheme shown in figure 1.

* This work has been funded by the Spanish CICYT contracts TIC2002-04491-C01/02 and INDRA S.A.
In section 2 an analysis of Madrid-Barajas airport layout is presented, where real sensors deployed are shown and the adopted ASTERIX format is presented. Section 3 describes the centralized fusion architecture designed. Section 4 presents some representative results of system behavior using several scenarios. Finally, paper conclusions are presented in section 5.

2 Airport Description: layout and deployed sensors

2.1 Airport Layout Original Data Base

The airport layout is stored in a data base that contains all the necessary information to define the physical layout and the arrival/departure routes defined by operator. The tables and their attributes are defined in the following paragraphs:

[Absolute Ground Points]
1. N° (Max. 1000). Point counter.
3. Point Type (Taxiway, RwyEnd, TaxiLane, Stand).
4. X Local Coordinates in cm.
5. Y Local Coordinates in cm.

[Absolute Segments]
1. N° (Max. 1000). Segment Counter.
3. Segm.Type (Runway, Taxiway, TaxiLane, Stand).
4. Points A. Point segment.
5. Points B. Point segment.

[Intersection Ground Points]
1. N° (Max. 500). Intersection Point Counter.
2. Point ID (15 characters) Intersection Point Id.
3. Point Type (Runway, TaxiLane)
4. Segment ID. Identification of the segment that contains the intersection point.
5. Refer. Point ID. Reference point for calculating the absolute values of the intersection point.
6. Distance in cm. Distance from reference point.

[Intersection Segments]
1. N° (Max. 500). Intersection Segment Counter.
3. Segm.Type (Taxiway, Standlane)
4. Points A. Absolute ground point.
5. Points B. Intersection ground point.

[Runways]
1. N° (Max. 12). Runways Counter.
2. Rwy ID (3 char.). Runway Identification.
3. Is heliport? (Y/N)
4. Threshold Point ID. Point Identification.
5. Segment ID. Segment Identification that defines the runway.
6. Touch Down Distance in cm. 4 cm.
8. First Entry Twy ID. Segment Identification of Rapid Entry.
9. Hook? (Y/N)

[Taxiways]
1. N° (Max. 100). Taxiways Counter.
2. Taxiway ID (8 char.). Taxiway Identification.
3. Segm. Sequence. Sequence of Segment Identification that define the taxiway.

[Taxilanes]
1. N° (Max. 50). Taxilane Counter.
2. Taxilane ID (8 char.). Taxilane Identification.
3. Segm. Sequence. Sequence of Segment Identification that define the taxilane.

[Standlanes]
1. N° (Max. 500). Standlane Counter.
2. Standlane ID (8 char.). Standlane Identification.

[Departure Ground Routes]
1. N° (Max. 500). Departure Routes Counter.
2. ROUTE ID (15 char.). Departure Route Identification.
3. INITIAL TAXILANE ID. Initial Taxilane Identification.
4. DEP RUNWAY ID. Departure Runway Identification.
5. INTERMEDIATE TAXI SEQUENCE. Intermediate Taxilanes identification.

[Arrival Ground Routes]
1. N° (Max. 500) Arrival Route Counter.
2. ROUTE ID (15 char.). Arrival Route Identification.
3. ARR RUNWAY ID. Arrival Runway Identification.
4. END TAXILANE ID. End Taxilane Identification.
5. INTERMEDIATE TAXI SEQUENCE. Intermediate Taxilanes identification.
2.2 Airport Layout Original Data Transformation and Data Base Storage for Fusion System

The airport layout ought to be defined by absolute points and absolute segments to be used by the IMM filter in the fusion process. Then, the first data transformation converts the intersection segment table in an equivalent table where each segment is defined by two absolute points. The original representation considers absolute ground segments composed by two absolute ground points and the intersection segment composed by one absolute ground point and one intersection ground point, as can be seen in Figure 2.

![Figure 2: Original segment representation in Airport Layout Data Base.](image)

The intersection ground point of intersection segment A could be transformed using the data base information. If intersection segment is named A, and it intersects with absolute segment B, we know an absolute point of B that could be used as reference and the distance to this point, as could be seen in Figure 3. Then, intersection point is transformed in an absolute point and the intersection segment in absolute segment.

![Figure 3: Transformation of intersection point in absolute point.](image)

The second transformation has been to decompose the runway description in a set of absolute segment. A runway is defined by two points and several taxiways intersect on it. In order to be used by our IMM filter, we need to obtain all the absolute points that define the intersections and the set of absolute segment that composed the runway. Once the set of segment has been obtained, we need to define the runway as a collection of segments instead of two points.

All the information obtained is stored in a relational data base where the integrity rules assure the comprehensive and coherent of the data stored. So, the segment table is related with the point table and if any segment is defined with a point that is not contained in the segment table the Data Based Management advises to avoid future problems. In the same way, taxilanes, taxiways and standlanes must be defined using the defined segments in the segment table. And, finally, runways must be defined using the segments stored in the segment table. The Data Base has been created with Microsoft Access, tables and relations appear in Figure 4.

![Figure 4: Relational Data Base based on table definition of airport layout.](image)

From the data base created, we obtain three text files to configure the simulator. The first file contains all the absolute ground points (the original ground points and the intersection ground points that have been transformed). The second file contains all the segments of the airport defined by two absolute points. The third file contains all the taxis (the original taxilanes, standlanes, taxiways and the transformed runways) to use the route definition stored in the departure/arrival ground routes tables. The Airport Layout built with this data is shown in Figure 5.
Figure 5: Madrid-Barajas Airport Layout.
2.3 Deployed Sensors and Asterix Format

The targets moving on the airport are general and commercial aviation aircraft, and surface mobiles, such as fuel trucks, baggage conveyors, 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.

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, Millimetre Wave Sensors. 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>
<thead>
<tr>
<th>Sensor</th>
<th>Cooperative</th>
<th>Id.</th>
<th>Mobiles</th>
<th>Meteorological</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMR</td>
<td>No</td>
<td>no</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>DGPS</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>all</td>
</tr>
<tr>
<td>MS</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>All</td>
</tr>
<tr>
<td>ADS-B</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>All</td>
</tr>
<tr>
<td>MWS</td>
<td>No</td>
<td>no</td>
<td>all</td>
<td>clear</td>
</tr>
</tbody>
</table>

In this work the types of sensors considered are: ASDE radar, ASR radar, S-Mode multilateration systems, ADS-B and Millimetre Wave Sensors. These sensors are deployed in Spanish airports and, then, they ought to follow the ASTERIX format. 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 [12], 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 a scheme that allows us to design different fusion architectures (centralized, distributed and hybrid) using the data received in ASTERIX format [11]. The main goal of the scheme is to manage the data received in ASTERIX format joint to sensor specifications in order to improve the local information received. Figure 6 shows this scheme.

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, Millimetre Wave Sensors. 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.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Cooperative</th>
<th>Id.</th>
<th>Mobiles</th>
<th>Meteorological</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMR</td>
<td>No</td>
<td>no</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>DGPS</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>all</td>
</tr>
<tr>
<td>MS</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>All</td>
</tr>
<tr>
<td>ADS-B</td>
<td>Yes</td>
<td>yes</td>
<td>equipped</td>
<td>All</td>
</tr>
<tr>
<td>MWS</td>
<td>No</td>
<td>no</td>
<td>all</td>
<td>clear</td>
</tr>
</tbody>
</table>

In this way, we have developed a scheme that allows us to design different fusion architectures (centralized, distributed and hybrid) using the data received in ASTERIX format [11]. The main goal of the scheme is to manage the data received in ASTERIX format joint to sensor specifications in order to improve the local information received. Figure 6 shows this scheme.

Fusion architectures could be developed using this general scheme. So, in [10] we describe several architectures, where 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. In [11], we analyze the performance of different architectures and the best performance is obtained using the centralized one.

The proposed centralized 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.

3 Fusion Architecture
The centralized architecture maintains a set of central tracks, \( \{T_i\} \). The measures received from ASTERIX sensors, \( \{P_{Skj}\} \), are associated to the central tracks and, then, filtered to actualize the track. In this architecture, the processes (association and filtering) work 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 7, the main processes of the centralized architecture are shown.

![Figure 7. Centralized architecture](image)

The centralized architecture is composed of the following steps:

1. **Coordinate Transformation**: All ASTERIX format measures, \( \{P_{Skj}\} \), with coordinate values respect to sensor position, \( \{S_k\} \), are transformed to unify the coordinates values with respect to the same global position, \( C \), \( \{P_{Cj}\} \).

2. **Temporal and Kinematic Compatibility**: Calculated for each pair measure-central track, \( \{P_{Cj}, T_i\} \). Gating function evaluates the possibility of plot-to-central track association.

3. **Association**: 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 \( \{P_{Cj}, T_i\} \) 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 or fused similar new tracks) is the final step.

The advantages of centralized architecture are: (a) Optimize the position estimation for any sensor measure variance, (b) minimize the effects of a delay between the time when a maneuver begins and when it is detected, because maximize the refresh rate. 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.

The software developed store the central tracks and the received plots (preprocessed but not assigned to any track) in a list. The track contains position and velocity of each target, filter parameters, identification and a list with the sensor used to actualize it (contains the last time of received plot from this sensor and the sensor identification of the track if available). This list is used in gating and association procedures. Besides, the track contains the sensor model, corresponding to each sensor, to geometrically transform the plot if needed.

Figure 7 represents the centralized model. The input data is the multisensor plots from the communications lines and a description of sensor deployed and airport layout. Each plot (measure) is preprocessed depending on the position of sensor that generates it. Then, the association process decides if the plot is used to actualize or to initialize a track. The output of the system is the track table, that is actualize with the developed filter (an IMM improved with the airport map).

![Figure 7: Data Structures in Centralized Data Fusion Architecture](image)
The list of central datum stored the pre-processed data that are waiting: (a) to be associated to existing track, (b) to initialize a new track, or, (c) to be deleted. The list of central tracks represents the information related with the existed targets: (a) estimated vector, (b) filter structures and (c) a list contained the information of all sensors that illuminate the target). In the list of sensor, the last illumination instant is stored to be used for temporal gating, and the code from each sensor. The system also includes a list of sensor models needed for transforming plots and the airport layout for filtering purposes. All the information sent from the sensor is referred to the sensor and must be transformed to be referred to fusion centre.

The fusion centre received periodically each T sec. the detected data from sensors. Each plot is transformed into a central datum following the sensor model that generates it. In each fusion cycle, the system, using the central tracks, execute the gating process, search for associations between plots and tracks, managing the list of pre-processed plots (received in this period) and the non-assigned plots of previous periods. If there are plots non assigned the initialization procedure try to create new tracks that must be confirmed later to be considered as central tracks. The whole process is represented in Figure 8.

In the figure 8, the estimation of bias is represented. This system maintains a set of estimators of systemic errors that appear in the sensor measures. These errors could be classified in two types: global and local. The global errors affect in the same way to all measures; the estimation of each error could be used to correct directly the processed measures. The local errors depend of the associated track; the estimation could be used after the association process.

![Figure 8: Process Diagram of Fusion Architecture](image)

**4 Conflict Detection System**

Our A-SMGCS prototype not only includes the surveillance function but it also comprises a basic conflict detection subsystem. Several kinds of conflicts are addressed in a centralized way, trying to assess the controller with alerts of hazardous situations. Our prototype has several cooperating conflict detections capable of detecting:

- Incorrect Landing: Madrid-Barajas Airport has two pairs of parallel runways. Therefore, a mistake in the landing procedure leading to the wrong runway is feasible. The conflict alert function assesses landing trying to confirm it is performed in the correct runway.
- Runway Incursion: This kind of conflict, either during landing or during taking off, is one of the most dangerous situations along flight. Our system is based on the activation of runways when it is being used for landing (aircraft aligned with runway, following glide path) or traversing it at high speed (either after landing or to take-off), and the detection of any aircraft entering a protected area if the runway is activated. The detection of runway activation processes through
artificial intelligence methods is being studied, trying to anticipate this activation as much as possible to reduce risk.

- Ground Movement Plan Conflict: aircraft, in a so congested and complex environment, should accurately follow an accorded surface movement plan. The conformance with this plan is assessed by this function. This function uses the routes, ways and segments described in section 2.1, tracking the conformance of aircraft movements with the assigned route.

- Non Authorized Movement: aircraft should follow accorded plans. But sometimes they break these plans, or there could be situations in which an aircraft has not a plan. Under these circumstances, the system assesses situations in which the aircraft follow, at least, not problematic routes. These routes are included in the airport database, and if the aircraft does not follow, at least, one of these routes, a conflict alert is raised.

- Separation conflict: Some conflict situations, in which the aircraft may approach too much to others during taxi movements, can be assessed by geometrical inspection of their relative situations, velocity, way they are following, etc. The problem of such a conflict detector is aircraft, in certain areas, tend to follow quite close paths. We are investigating in our prototype the use of expert systems to assess these conflicts, and the potential training of such detectors.

All these conflict situations and associated conflict detectors make use of the airport database and the fusion derived tracks. They perform conflict assessments in real time, and so fast access to database is necessary. This lead to an object oriented design of the airport database with pointer to access all necessary information. This design is depicted in figure 9.

![Figure 9: Airport Database Design](image)

In this figure, continuous line has to do with “has-a” and “has-several” relations, and discontinuous lines are fast access links, which in some cases may even be not present (pointed to NULL). This entire database is compiled during conflict detection system initialization. Conflict detection is then based on the localization of tracks on segments. Then, conflict track, in the localization phase, are filled with segment, runway, potential routes, and runway access pointers, to be used in all later detectors, using the fast access links described.

5. Conclusions

Centralized fusion architecture has been developed 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. Besides, the A-SMGCS system is composed of a conflict detection module that alerts when predicts an unsafe situation. The deployed sensors follow the ASTERIX normative [13], 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.

References