Abstract - The work presented here addresses practical aspects of data fusion to implement a prototype of Advanced Surface Movement Guidance and Control Systems (A-SMGCS). It reflects recent experiences in an on-going project to fuse data from available and future sensors at Madrid-Barajas airport: Automatic Surface Detection Equipment (ASDE), Millimeter Wave Sensor (MWS), Airport Surveillance Radar (ASR) and Mode-S multilateration. Simulation results show system accuracy and robustness in representative situations, taking into account the airport configuration.

Keywords: ASMGCS (Airport Surveillance), Tracking, Data Sensor Integration

1 Introduction

In this work, the core data fusion algorithms designed for a real A-SMGCS prototype, to be installed at Madrid-Barajas international airport [1], are described. They are integrated in a centralized architecture, maintaining a list of central tracks updated by multi-sensor data received from installed sensors, using ASTERIX protocol for airport data interchange [2]. Consequently, there are centralized processes for data association and filtering, performed at measurement level, taking into account airport information such as map representation, and sensor characteristics. The data fusion output is intended to be used in the higher-level monitoring function, in the conflict detection process [1].

The Surveillance function will detect and track all targets moving on the airport: commercial aviation aircraft and surface vehicles such as follow-me cars, fuel trucks or luggage convoys. The sensors are basically classified into cooperative, which provide identification but require active cooperation from targets, and non-cooperative, which provide global detection capability but lack identification. In the second group, the ASDE [3] is one of the most representative data sources, although the use of short-range radar networks, such as Millimeter Wave Sensors (MWS), is an attractive solution [4]. Regarding cooperative sensors, we can mention surface multilateration systems, such as Mode-S multilateration, or systems based on D-GPS+ADS-B [5]. Therefore, the Surveillance function must integrate data from these different sensor technologies into an unified representation.

2 High-level algorithms structure

The structure of data fusion algorithms is outlined here, taking apart the registration algorithms, which will be detailed in [6]. Data input are multisensor measurements received by means of ASTERIX protocol, a representation of available sensors and the airport map:

Data input: received data, sensorList, airportMap
States: trackList
Internal variables: dataList, associationPairs

For each processing time (period of T seconds)
- Read sensor data in last interval: \( n \cdot T < t < (n+1)T \) and apply coordinate transformation: dataList
- Apply Association algorithms:
  associationPairs = Associate(dataList, trackList)
- Apply Tracking Filter algorithm:
  trackList = Update(associationPairs, trackList)
- Apply Initiation/Deletion algorithm:
  trackList = Management(dataList, trackList)

All data received from sensors in the airport communication network are buffered. This is so because the data fusion is an independent process working in parallel with sensor network. With a constant period \( T (0.8 \text{ seconds}) \), the data fusion process is invoked and, in the first place, it reads the new received data to create the central data list, dataList, applying coordinate transformation. The latency of 0.8 seconds is acceptable since it is required. Then, these data are associated in order to update current tracks in trackList, in the Update function, or used to initialize new ones in the Management function, depending on the decisions taken in the Associate function.

---

* This work has been funded by INDRA, the Spanish CICYT (TIC2002-04491-C01/02) and CAM (07T/0034/2003 1)
3 Design of Data Association

The association process is based on three elements which take into account specific characteristics of this application: tracks-to-data gating to discard incompatible pairs, code-based assignment to perform direct assignments, and minimum-distance assignment. The first association steps, gating and code-based assignment, allow a reduction of the assignment table size, in order to increase efficiency. The last element is the real data association core, which evaluates a 2D distance for all possible pairs not previously associated. The process timing is depicted next.

Due to the quality of airport sensors, the association scheme is a simple sequential process, where every data is associated when it is received, without opening hypotheses as in more complex alternatives such as MHT tracking [7]. However, errors due to a hurried assignment of recent data are avoided with an additional delay, a “guard time” $\Delta T_{as}$. This time deferral means that data received with an age lower than this parameter ($t_{\text{Datum}} > t_{\text{FusionTime}} - \Delta T_{as}$), are processed twice. The first time they go in the association process (to compute gates, distances, etc.), but their assignment is deferred to the next processing time, when more context information is available to take a safe decision.

3.1 Gating

The gating process carries out a time-space compatibility checking. In the case that this double test is not attained, the distance between track and datum is marked as “INF” (infinity), which means that this pair will not be considered in the distance-based association. Tests are:

- **Time compatibility** is done by comparing data acquisition time, $t_D$, with last update time for that track and that sensor, $t_{ts}$. Since we have multi-sensor tracks, the information of last update, for each active sensor updating this track, is stored. It is taken into account the sensor scan period, $T_s$, and its time jitter, $\sigma_S$. Besides, there can be some update losses, and they should not lead to automatic loss of following data, so the final time condition is

$$t_D \in [t_{ts} + NT_s \pm \sigma_S] \quad (1)$$

being $N$ any positive integer in $\{1, ..., 5\}$.

- **Space compatibility**. It is done by computing a distance between track and datum, considering that they could be have different dimensionalities:

  - if datum and track have the same dimensions or datum is 3D, the distance in horizontal plane is computed.
  - if datum is 2D and track 3D, datum is first corrected with track height (to avoid slant-range effects with Surface Radar), and then the distance in horizontal plane is computed.

The practical space gates implemented are rectangular, in order to easily parameterize the maximum allowed maneuvers, depending on the area where the target is located.

$$|\xi_{\text{Datum}} - \xi_{\text{Track}}| \leq 3\sigma_{\xi} + \frac{1}{2} a_{\xi-\text{MAX}} \cdot t_{\text{delay}}^2 \quad (2)$$

where $\xi$ is a horizontal coordinate; $\sigma_{\xi}$ the residual standard deviation on that coordinate; $a_{\xi-\text{MAX}}$ is the maximum expected maneuver projected along $\xi$ (depending on the airport area) and $t_{\text{delay}}$ is the time reaction of tracking filter in a maneuver transitions.

3.2 Code-based association
The code availability is used to improve efficiency with direct assignments [10]. Since it is a centralized architecture, the heterogeneous code systems (SSR codes, Mode-S, ICAO address, etc.) are integrated in every track accordingly to the active updating sensors. Direct assignment is allowed when datum and track (for that specific sensor) have the same code. Besides, the following conditions are checked:

- Time-space gating has been successful.
- Age of datum to assign is higher than guard time, \(\Delta T_{as}\).
- Track’s ID code has achieved the “stable” state.

Given that all conditions are attained, the direct assignment is carried out. Otherwise, the datum will go through the distance association algorithm. The last condition, a stable code, is used since sometimes errors or switches in received codes might appear. Basically, the condition to be in “stable-code” condition is receiving three consecutive identical identifiers assigned to the track. When a datum with different code is assigned, track enters in code-change state, waiting for three consecutive codes. However, if during this situation the original code is received, it goes back to code-stable situation. This logic is implemented using, for each active sensor, counters for the number of old and new codes received \((\text{countID}, \text{countN})\), and their code values, \((\text{codeID}, \text{codeN})\). The code of current datum is \(\text{codeR}\). The process is depicted in figure 2.

3.3 Distance Data Assignment

Finally, remaining data are associated to compatible tracks, computing the pairs with minimum distance in the assignment table. This comparison takes into account local corrections such as height completion for 2D data or local biases removing [6]. Since assignment conflicts appear only among data from the same sensor, a different distance table is computed for each active sensor. Data in \(\text{dataList}\), with the exception of those assigned by code in the previous step, are compared with all compatible tracks. For those tracks, which are not compatible with that datum, a label INF is set in the table. An example can be seen in figure 3 and the associated distances are presented in table 1. There are four tracks, and only data gated from two sets tracks pairs have association conflicts: data \(D_1, D_2\) to tracks \(T_1, T_2\), and \(D_4, D_5\) to \(T_3, T_4\). In the operation with all data in airport surface, the table built has many incompatible pairs. The assignment algorithm (Munkres [8]), is efficient to take this fact into account and solve the global problem.
Table 17: Distances for all datum-track pairs

<table>
<thead>
<tr>
<th>d_{ij}</th>
<th>D_1</th>
<th>D_2</th>
<th>D_3</th>
<th>D_4</th>
<th>D_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_1</td>
<td>d_{11}</td>
<td>d_{12}</td>
<td>d_{13}</td>
<td>INF</td>
<td>INF</td>
</tr>
<tr>
<td>T_2</td>
<td>d_{21}</td>
<td>d_{22}</td>
<td>d_{23}</td>
<td>INF</td>
<td>INF</td>
</tr>
<tr>
<td>T_3</td>
<td>INF</td>
<td>INF</td>
<td>INF</td>
<td>d_{34}</td>
<td>d_{35}</td>
</tr>
<tr>
<td>T_4</td>
<td>INF</td>
<td>INF</td>
<td>INF</td>
<td>d_{44}</td>
<td>d_{45}</td>
</tr>
</tbody>
</table>

4 Design of Tracking Filter

Regarding the tracking filter design, it can be seen as a particular case of Ground Target Tracking, a field which has recently received quite attention in tracking algorithms literature [9]. The interesting objects move accordingly to physical shapes of airport roads, reflected in the map, and follow certain procedures (types of motion patterns, acceleration magnitudes, etc.), so the tracking procedures can be different from those used in other free environments. The implemented solution is an IMM filter with modes specifically designed to exploit the available surface information. The proposed filter is composed of three modes (see figure 4): constant speed motion, aligned with the segment orientation where target is currently located; segment-transition filter, performing transversal manoeuvres in order to move to close segments when the target crosses branching areas; and longitudinal-transition filter, adapting to stop and go, or runway acceleration manoeuvres. The IMM structure is displayed in figure 7.

From this list, the closest segment to current position is selected to define orientation of principal mode (mode 1) and the second one to define the adjacent mode (mode 2). This information of segment orientation is directly integrated in the prediction structures for motion prediction, following the equations proposed in [9]. They provide a considerable improvement in the accuracy of the estimates, due to continuous correction on heading which leads to good smoothing in segments at constant velocity. Besides, the response to maneuvers is faster for transversal ones, when the target maneuvers to keep the trajectory inside the boundaries of the road. The mode 2 allows quick transitions in branching points, as depicted in next figure, while mode 3 is intended to improve performance under longitudinal manoeuvres such as acceleration in runways or stop-and-go motion.

Therefore, the first step in the filter structure (figure 7) is the location of targets on airport structure [1], determining the closest segments to current measurement, ranked by their transversal distances.

A practical aspect is the introduction of a specific mode for transitions (see figure 6). This mode takes the angle of closest segment to current one, and a higher variance. However, when a branching effectively takes place and IMM transition is carried out, the location now will be done over map segment which was previously mode 2, and the closest segment will be the mode previously corresponding to mode 1. From that moment on, it is not applicable the classification before. Modes should switch to keep the state and probability computed for mode 2 when it is labeled as mode 1. The switching logic identifies this situation of branch to closest segment. A double condition is checked: first, the probability of mode 2 must be higher than probability of mode 1. Secondly, at least two consecutive
locations must be done in the same segment, the one stored as closest segment in last update.

\[
\begin{align*}
\hat{x}_3[k] & \quad \hat{P}_3[k] \\
\hat{x}_2[k] & \quad \hat{P}_2[k] \\
\hat{x}_1[k] & \quad \hat{P}_1[k] \\
\end{align*}
\]

\[
\begin{align*}
\Sigma_1,k & \quad \Sigma_2,k & \quad \Sigma_3,k \\
\end{align*}
\]

Figure 7. IMM Filter for track update in airport surface

5 Results

Here, an illustrative evaluation of the proposed algorithms is presented, referred to some representative scenarios simulated on Madrid-Barajas airport. To do that, simulation and evaluation tools have been applied to the whole implementation of data fusion architecture and algorithms, using the specific airport ways and operational routes, and simulated models of available and to-be-installed sensors. Figure 8 depicts the main components of simulated environment, airport map and deployed sensors. The airport has four runways with the corresponding access taxiways and the internal taxi network connecting them with four aprons: terminals T1, T2, T3, X. An apron is the airport area including parking positions, and zones for passenger embarking and loading cargo.

The available sensors are ten: two ASDE, one approach secondary radar (ASR), a multilateration Mode-S network (MS), with four installed stations, and six millimeter wave sensors (MWS) deployed in one apron. Their characteristics are summarized in table 2.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Position</th>
<th>T_{\text{scan}}</th>
<th>Error</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASDE1</td>
<td>(-900, 400) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>5 Km</td>
</tr>
<tr>
<td>ASDE2</td>
<td>(-600, 3700) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>5 Km</td>
</tr>
<tr>
<td>Multi-Mode-S</td>
<td>Four stations: $((-4, -2), (3, -2), (3,7), (-4,7))$ Km</td>
<td>1 s</td>
<td>$\sigma_x=5m$, $\sigma_y=5m$</td>
<td>Quadruple coverage in surface</td>
</tr>
<tr>
<td>ASR</td>
<td>(10000, 0) m</td>
<td>4.8 s</td>
<td>$\sigma_r=10m$, $\sigma_\theta=0.09^\circ$</td>
<td>100 Km</td>
</tr>
<tr>
<td>MWS1</td>
<td>(-2770, 1550) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
<tr>
<td>MWS2</td>
<td>(-2770, 2850) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
<tr>
<td>MWS3</td>
<td>(-2400, 3030) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
<tr>
<td>MWS4</td>
<td>(-2100, 2550) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
<tr>
<td>MWS5</td>
<td>(-2100, 2025) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
<tr>
<td>MWS6</td>
<td>(-2100, 1650) m</td>
<td>1 s</td>
<td>$\sigma_r=5m$, $\sigma_\theta=0.15^\circ$</td>
<td>600 m</td>
</tr>
</tbody>
</table>
A set of operational surface routes have been selected for simulation, defining trajectories following these routes for landing and take-off. Besides, in order to analyze the robustness when distances are reduced, there have been added vehicle trajectories close to aircraft. In next figure, some simulated trajectories for take-offs are displayed:

There are figures for three different situations of available sensors: only ASDE and ASR (current operative sensors), addition of MS system, and addition of MWSs. Position errors vary between 1 and 4 m in uniform motion, and maneuver peaks are around 8 and 12 m. The situation of longitudinal acceleration does not produce important effects in position, with the exception of velocity groundspeed. Generally, it can be seen a significant improvement with the introduction of new sensors, especially in the smoothing at uniform motion. This is mainly due to the higher update rate, which makes IMM have lower transition probabilities (they depend exponentially on the time elapsed between consecutive updates).

5.1 Accuracy results

As an example of accuracy results for a representative situation, figure 10 presents the RMS values of position and velocity errors for a departing aircraft on runway RW18R.
5.2 Continuity results

The continuity performance with previous experiments was always satisfactory (a single, continuous track per real target). In order to demonstrate system robustness, two scenarios with harder conditions were simulated, one with very close separation distances, below the allowed separations, and other with a very high rate of false alarms. In the first case, several vehicles perform sharp maneuvers (accelerations up to 20 m/s²) moving very close to an aircraft. The trajectories are displayed in figure 12, while separations among all objects are displayed in figure 13. It can be seen that minimum distances are around 25-30 m.
and ASR radars, and adding the MS sensor. There is a double advantage in the use of this cooperative sensor. First, a reduction of switches from 4% to 2%. Second, track switches only appear in one sample (the situation is recovered later) while permanent switches appear in the case with only primary radars.

The second situation analyzed is the effect of false alarms, coming from a bad adjustment of ASDE radars. A landing trajectory was used (runway RW18R), with clutter regions located at two maneuvering areas: a rapid exitway, and a 180° turn. So, false alarms are located in an especially sensitive area, and their expected effect is maximum. An example is depicted in figure 15.

The second situation analyzed is the effect of false alarms, coming from a bad adjustment of ASDE radars. A landing trajectory was used (runway RW18R), with clutter regions located at two maneuvering areas: a rapid exitway, and a 180° turn. So, false alarms are located in an especially sensitive area, and their expected effect is maximum. An example is depicted in figure 15.

As in the previous results, a comparison is done between the use of current sensors (two ASDEs and on ASR), and the installation of Mode-S multilateration (MS). Besides, the probability of false alarm (per spatial cell) took values \{0.001%, 0.01%, 0.1%, 1%, 2\%\}. The switching rate is presented in figure 16. The system robustness guarantees no effect until 0.01% of false alarms. The effect is very heavy from a rate of 1%. As in other cases, the advantage of including the cooperative sensor is very significant. The switch probability falls from 85% to 2% with FA rate of 1%.

6. Conclusions

This work presented some practical details of data fusion algorithms implemented in an A-SMGCS prototype, intended to operate in Madrid-Barajas airport. The analysis of results shows the satisfactory performance of designed system, in terms of accuracy and robustness. Realistic conditions close to real-world operations have been simulated, and harder conditions to analyze the system response. The main conclusions of the evaluations are:

- The effect of including additional sensors is very significant, especially if they are cooperative and provide identification. The improvement in accuracy is basically the higher smoothing factor, although the system could be configured to translate this advantage to the maneuvers transitions.

- The use of map information is very positive, outstanding the improvement in transversal and heading accuracy.

- The continuity was totally satisfactory in realistic conditions. Under very hard conditions (low separations and high level of false alarms), the inclusion of cooperative sensors to increase tracking robustness was important. In any case, there were track losses/switches under conditions of separations below of 20m and maneuvers higher than \(20 \text{ m/s}^2\), or false alarm rates higher than 0.1%.

References


