Sensor Modeling and Data Processing for Airport Simulation

Antonello Pasquarelli
ALENIA MARCONI Systems
Via Tiburtina km 12.400, 00131 Rome – Italy
apasquarelli@aleniasystems.finmeccanica.it

Andrea Corsini, Andrea Garzelli
Department of Information Engineering
University of Siena, 53100 Siena – Italy
{corsini,garzelli}@unisi.it

Abstract - This paper presents the SEEDS simulation environment for the evaluation of distributed traffic control systems. The description starts with a general overview of the simulator, targeted for airport surface traffic simulation, and then focuses on the sensor models implemented in the prototype. The surveillance function foreseen in a real Advanced Surface Movement Guidance and Control Systems (A-SMGCS) has been studied and modeled; suitable set of sensors and signal processing algorithms have been considered and their performances have been analyzed in order to be compliant with the application performance requirements defined by International Organizations. The paper shows the sensor module architecture, how the sensors have been modeled and how the software module has been implemented and integrated in the core simulator. The interactions with the other modules of the simulator and the exchanged messages are also described.

Keywords: Sensor modeling, simulation, airport surface traffic, advanced surface movement guidance and control systems (A-SMGCS).

1 SEEDS Overview

SEEDS (Simulation Environment for the Evaluation of Distributed traffic control Systems) is a distributed HPCN simulation environment composed of powerful workstations connected in a local network and it is targeted to the evaluation of Advanced Surface Movement Guidance and Control Systems (A-SMGCS). The simulation environment allows the definition and evaluation of technologies and performances needed to implement new functions and procedures of A-SMGCS, to mould new roles in the airport, to introduce new automatic tools and interfaces, to support A-SMGCS operator decisions.

The SEEDS consortium is composed of Alenia Marconi Systems (I), as coordinator, Sogitec (F), Artec (B), as industrial partners, University of Siena (I), LRR-TUM at Technische Universität München (D), Slovak Academy of Science (SK), as associated partners, and Sicta (I) as final user-partner. An European User Group, composed by the flight assistance administrations Sogel (L) and SEA (I), participated to all the phases of the project.

The software architecture of the simulator, defined using the Unified Modeling Language (UML) notation, is based on CORBA (Common Object Request Broker Architecture) as communication middleware; the DIS (Distributed Interactive Simulation) protocol has been used for the image generation and distribution. This choice assures the scalability of the system, allowing the mapping of the different objects on heterogeneous workstations or PCs. Techniques for load balancing have been used to reduce computing power, data rate and latency on the network. In particular centralized versus distributed architecture for traffic and image generation have been analyzed, and the distributed solution was adopted in SEEDS for performance reasons. Subscription and notification mechanisms have been used to synchronize processes in the distributed environment.

The simulation environment is composed of commercial off-the-shelf components and of some proprietary software modules, and it is open to be connected to other ATM (Air Traffic Management) simulators. The main software modules of the SEEDS architecture, completely developed in C++ and reported in Fig.1, are: Scenario Generation and 2D-3D Visualization; Sensor Models; Airport Database; Surveillance, Control, Guidance and Planning Modules; Controller, Pilot and Driver models; Administration Station modules.

The scenario generator is in charge to animate the scene according to the stimuli coming from the actors of
the simulation and from the External World. The actors can be simulated (software processes) or real (human beings). The 3D visualization reproduces the scene as seen by the actors’ eye (eye model); the 2D visualization reproduces the scene as seen by the sensors present in the airport (ASDE radar, GPS, DGPS, Magnetic/Dynamic sensor error models). The controllers have a set of DSS (Decision Support Tools) which help them to plan the aircraft surface movements. The other functions of A-SMGCS (Surveillance, Control and Guidance) are also implemented. An Administration Station is responsible to configure, start-up, stop, restart the simulation, and it collects application level and system level statistics.

2 Sensor Modeling

This paragraph describes the two main sensors used in an A-SMGCS environment and considered in the simulation environment: the radar and the global positioning system. The sensors’ principal parameters and performance are studied in order to build suitable mathematical models which reproduce their behavior in the simulated environment. This models receive as input all parameters needed to characterize each sensor and the environment, and produce as output the measures with the same resolution and accuracy obtainable from the real sensors.

The sensor models have been implemented as on-line algorithms taking into account the performance requirements of the real time simulation.

In the simulated environment, sensor models are used to present the traffic as perceived by these sensors, according to the operational characteristics of the available sensor systems, and to the external world (weather conditions). The changes in the meteorological situation on the airport can be accomplished by storing all changes and operations into a relational database in order to perform either on-line or off-line data processing and fusion. In fact, data processing algorithms can operate according to two approaches: on-line approach and off-line approach. In the on-line approach, the algorithms are studied and implemented as adherent as possible to the real system: for example, considering tracking algorithms, they receive in input data coming from the sensor error models and process the data by means of Kalman filter or $\alpha$-$\beta$ filter.

In the off-line approach, sensor integration is modeled by means of statistical parameters. Algorithm performances are globally characterized by statistical figures, such as mean and variance, referring to a probability density.

The movements of mobile objects (aircraft and vehicles) are generated according to their cinematic characteristics and to the information contained on the Flight Data List producing the mobile object ground truth. Then traffic is presented according to the models of the sensors used at airports and to the weather conditions, producing the mobile object perceived truth for 2D and 3D visualization (Fig.2).

![Fig.2 Example of synthetic image](image)

2.1 ASDE Model

The main sensor currently used in airport surface surveillance systems is the Airport Surface Detection Equipment (ASDE) with its current version ASDE-3. It consists of a radar-based detector, display processor, display subsystem, and other ancillary subsystems.

The ASDE model which has been developed is based on the simplifying assumption of a free-space model, i.e. not considering diffraction phenomena, while the reflection effect of the terrain surface has been taken into account. The assumption of free-space model is satisfied with the hypothesis of high-directivity antennas.

In the following table, an analysis of the characteristics and obtainable performance of a commercial ASDE radar (by Cardion) is reported.

<table>
<thead>
<tr>
<th>Table 1 Characteristics of a typical ASDE radar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
</tr>
<tr>
<td><strong>PRF</strong></td>
</tr>
<tr>
<td><strong>Pulse length</strong></td>
</tr>
<tr>
<td><strong>Azimuth beamwidth</strong></td>
</tr>
<tr>
<td><strong>Elevation coverage</strong></td>
</tr>
<tr>
<td><strong>Radar Update Rate</strong></td>
</tr>
<tr>
<td><strong>Polarization</strong></td>
</tr>
<tr>
<td><strong>Range of Display</strong></td>
</tr>
</tbody>
</table>

Among all factors affecting the measurement of the target position in polar coordinates, a subset of parameters have been considered, due to the hypothesis of free-space model.

In particular, the following parameters have been considered for ASDE modeling:
- Radar Pulse Width, affecting the range estimate;
- Atmospheric Attenuation, affecting the range estimate and depending on radar frequencies, elevation angle, rainfall rate, clouds and fog density;
- Antenna Radiation Pattern, affecting the angle estimate;
- Surface reflection effects, affecting the angle estimate and depending on the half-power beamwidth elevation angle;
- Surface reflection coefficients:
  - Fresnel reflection coefficient (depending on different polarization wave);
  - Specular scattering coefficient;
- Target elevation;
- Grazing angle;
- Path length difference;
- Range to reflection point;
- Target height.

The radar model implementation has been validated by comparing it to the radar model developed by Schwab & Rost [6]; they presented a software implementation based on Cardion ASDE-3 technical characteristics and made comparison between predicted and measured results having a good correlation (less than 5 dB difference between values at 1500 - 2600 meters range). In figure 3 we compare our prediction with the Schwab-Rost model. It has been made under the same restrictions (1.6 m$^2$ RCS, 16 mm/hr rainfall rate, single pulse) reported in [6]. It should be noted that, in the above-mentioned distance range, the SNR of the implemented ASDE model correlates the measured data as the Schwab-Rost model did with comparable performances.

The estimation of the target position by an ASDE radar is affected by several factors, as described above. The error affecting a given measure is expressed by either the standard deviation of a Gaussian probability density function or an absolute error. For both cases we propose to consider a simplified error model: we consider that the reference value is affected by Additive White Gaussian Noise having zero mean. We considered also the Detection Factor, a threshold value for target detectability given a received Signal to Noise Ratio. The values which have been used in the simulations are 0.9 and 10$^{-6}$, respectively, since they are typical in common ASDE radar systems.

Range errors have been evaluated differently from angle errors. In the first case, an interval between threshold-crossing of noise-affected pulse and of non-noise-affected pulse has been derived in the following equation:

$$
\delta T_r = \left( \frac{\tau}{4B E/N_o} \right)^{1/2}
$$

(1)

where $\delta T_r$ is Noise Added Pulse Length, $\tau$ is the pulse length, $B$ is the Intermediate Frequency (IF) bandwidth, and $E/N_o$ is the signal to thermal noise energy ratio.

The Noise Added Pulse Length can be multiplied by c/2 (c is the Light Speed) to give the standard deviation of the range measure.

Angle errors have been evaluated by the following equation:

$$
\delta \theta = \frac{0.628 \delta \theta_y}{\sqrt{2 E/N_o}}
$$

(2)

where $\delta \theta_y$ is the standard deviation of angle measure, $\theta_y$ is the half power beamwidth angle, $E/N_o = \text{Signal to Thermal Noise Energy Ratio}$.

Both values $\delta T_r$ and $\delta \theta$ have been used to evaluate the target error position. The ASDE SW Interface receives input from environmental parameters (like object characteristic and weather conditions) and system parameters (radar characteristics). To characterize the Signal-to-Noise-Ratio and the objects error positions, the following noise sources have been considered: thermal noise, ground clutter (due to the ground reflection), volume clutter (due to precipitation). Each contribution has been characterized by a mathematical equation. The Signal-to-Noise-Ratio has been derived considering contributions from signal power and noise. In particular, the classical radar equation for modeling the received signal power has been considered, and the volume clutter power, the ground clutter, the precipitation volume, and the precipitation volume reflectivity have been modeled.

Figures 4 and 5 show the results obtained for two radar configurations having different transmission frequencies and technical characteristics. Operating frequencies are 16 Ghz and 9 Ghz, and the elevation
beamwidths are 1.6° and 22°. These different characteristics cause different behaviors for varying weather conditions.

Elevation beamwidth strongly reduces performances in 9 GHz Radar System, especially in bad weather conditions.

2.2 GPS, DGPS Model

The Global Positioning System (GPS) and the Differential Global Positioning System (DGPS) have been studied and modeled in order to reproduce their behavior and performance in the simulation environment. The GPS and DGPS receiver models should be able to produce all data that can be experienced by a user equipped with a real GPS or DGPS receiver and with the same performance and accuracy.

The Standard Positioning Service has been taken into account and the performance standards relative to the constellation of satellites (i.e., Coverage, Service Availability and Service Reliability characteristics) are used to define a model of the satellite constellation. The model is built starting from results contained in [5] and it is a Prior Probability Model (PPM). This approach produces a model of the system failure which takes into account any prior probability information that might influence the effectiveness of the measure.

A spacecraft failure/renewal model has been assumed and failures have been divided into “hard” and “soft” ones: hard failures cause a complete satellite shutdown, soft failures cause a satellite operating with ranging signal errors. The basic parameters used to model the chosen failure/renewal model are the mean time between failures (MTBF) and the mean time to restore (MTTR). The model produces the coverage, service availability and service reliability, which are global parameters expressed as a probability; they model the availability of the system for reliable position measurements.

Service availability (Psa) is conditioned to coverage (Pc); this means that the service can be available when the coverage is assured. Service reliability (Ps) is conditioned to service availability and service reliability; this means that the service can be reliable when the coverage and service availability are assured.

Given coverage, service availability and service reliability as global conditioned probabilities, the GPS receiver model evaluates the error affecting the position measure using predictable and repeatable error characteristics expressed in terms of statistical distributions. The error in the position measurement can be modeled with Gaussian distributions with different mean and variance in North and East directions. Figure 6 reports the GPS receiver error model.

DGPS is based on the principle that receivers nearby positioned will simultaneously experience common errors on a particular satellite ranging signal. A typical DGPS architecture consists of a reference station located at a known location, and one or more DGPS user mobile receivers. The user receivers use measurements from the reference station to remove the common errors.

![Fig. 6. GPS Receiver Error Model](image)

The considerations about the GPS are also valid for DGPS, where an improved position measurement is obtained applying corrections from a reference station. A further parameter affecting the probability that a measure can be made, is the Reference Station Continuity (Pm) which is the probability that the reference station will be available for the duration of a phase of operation. Figure 7 reports the DGPS receiver error model.

![Fig. 7. DGPS Receiver Error Model](image)

3 Data Processing

3.1 Tracking algorithm

The Kalman filter has been adopted, in the case of three-state filter, for application to the SEEDS Project.

We assumed that the target dynamic process can be modeled in the discrete Markov form:

\[
x(k+1) = \Phi x(k) + q(k) + f(k+1|k)
\]

where \(x\) is the n-dimensional target state vector that includes the quantities to be estimated. Also, \(\Phi\) is the assumed known transition matrix; \(q(k)\) is the zero-mean, white, Gaussian process noise with assumed known covariance \(Q\); and \(f(k+1|k)\) is a known deterministic input, such as the relative position change associated with own-ship motion.

Measurements are in the form of linear combinations of the system state variables, corrupted by uncorrelated noise. Thus, the M-dimensional measurement vector is modeled as

\[
y(k) = H x(k) + v(k)
\]

where \(H\) is the M×n measurement matrix, and \(v(k)\) is zero-mean, white, Gaussian measurement noise with covariance \(R\).
Let $P$ (symmetric $n \times n$ matrix) denote the covariance matrix defined in terms of the zero-mean Gaussian estimation error vector. We considered $n=3$, $x=[x,v,a]$, $H=[1,0,0]$, and the transition matrix $\Phi$ has been assumed upper triangular with $\Phi_{11}=\Phi_{22}=\Phi_{33}=1$, $\Phi_{12}=\Phi_{23}=\Phi_{31}=T$, $\Phi_{13}=T^2/2$. Therefore, the Kalman gain equation gives

$$K_1 = \frac{p_{11}}{p_{11} + \sigma_n^2} \quad K_2 = \frac{p_{12}}{p_{11} + \sigma_n^2} \quad K_3 = \frac{p_{13}}{p_{11} + \sigma_n^2}$$

Thus, the first state gain ($K_1$) depends on the ratio of the estimation error variance to the total residual (including measurement noise and estimation error) variance. The other two gains ($K_2$, $K_3$) are computed through the covariances between the second and third states and the first state that is observed.

To initialize the filtering process, the following values have been assumed for the covariance matrix elements and for the variance of the acquisition noise:

$$p_{ij} = 0 \quad (i \neq j)$$

$$R_e = r_e \leq \sigma_n^2 \left( \frac{E_{\text{MAX}}}{3} \right)^2 = 2.77 \, m^2$$

$$p_{11} = \sigma_n^2 \quad p_{22} = \sigma_n^2 \quad p_{33} = \sigma_n^2$$

A software module implementing the above mentioned three-state Kalman filter has been developed and tested both for on-line and off-line processing in the SEEDS environment.

Worst-case situations have been taken into account:
1. Landing situation: an object moving along a trajectory divided into two sections:
   - a straight trajectory $t_1$ with 2 m/s$^{-2}$ acceleration and initial velocity 300 km/h;
   - a straight trajectory forming an angle of 60° with $t_1$ and constant velocity 70 km/h.
2. Take-off situation: an object moving on a straight, constant-acceleration trajectory.

For both situations a typical error variance reduction of 30% has been computed in terms of mean square error (MSE) with respect to the predefined trajectory.

3.1 Data fusion

The first step to integrate GPS and ASDE radar data is the gating process, that is the decision whether a new observation belongs to the track of a detected object or if it indicates the presence of a new target.

In the SEEDS project we adopted a technique using two levels of gating. The first gate is defined by means of the covariance matrix $P$ of the Kalman filter, while the second gate, usually for maneuvering targets, is based on a more detailed model of the possible object trajectories. The Wong and Korsak method has been used, which is based on the aircraft kinematics characteristics.

The following phase is the plot-track association. To limit the computational complexity, a sub-optimal algorithm of the nearest neighbor technique has been developed and tested.

4 Software Module Architecture

The Sensor Model and Data Fusion Module (SMDFM) has been implemented according to the Common Object Request Broker Architecture (CORBA) which is a standard for distributed, object-oriented client/server systems. It is defined and maintained by the Object Management Group (OMG).

The most important component of the CORBA middleware is the Object Request Broker (ORB). Its task is to provide the basic interaction capabilities that allow objects to communicate. The application objects make use of the ORB functionalities in order to realize distributed applications. Additional objects are given by the Object Services and the Common Facilities, which contain common and often required services that may be used by the application objects.

![Object Request Broker](Fig. 8. Object Request Broker)

When client objects want to make use of a specific server object, the ORB is responsible for enabling the communication between the objects. Figure 8 shows how the ORB is used to pass requests from a client to a server object represented by an object implementation. With this kind of communication, location and platform transparency is given because the client doesn’t need to know about the current location of the server’s object implementation and of respective system platform.

In a second step, the Object Request Broker (ORB) is defined within the CORBA specification. It specifies how objects may communicate with another using the ORB functionalities. In a CORBA environment, objects that want to provide specific functionalities (called services) have to define their interfaces using an Interface Definition Language (IDL). This language is part of the CORBA specification and allows to specify interfaces in a language-neutral way. An IDL compiler is used to map the interface specification to programming languages used for the client or server implementation.

![Component Diagram](Fig. 9. Component Diagram)
The interaction between SMDFM and other software modules which use data from sensors is based on CORBA. Figure 9 reports the component diagram restrict to the relationships between the SMDFM software module and other modules of the software architecture defined in [4].

Following the CORBA terminology, the SMDFM module takes the data needed for internal computation acting as a client and provides the interested modules with results acting as a server. In particular:
- The Administration Station is responsible for the server configuration at the start-up of the simulation session. It configures the type and number of sensors and their systems parameters.
- The External World provides the SMDFM with the meteo information, which affects the sensor performances, and Flight Data List information used for labeling.
- The Traffic Generator produces the mobile objects ground truth, i.e. real positions of the mobile objects.
- The SMDFM produces the system tracks, which are the mobile objects perceived truth.
- 2D View, Control Function, Guidance Function use the mobile object perceived truth, i.e. the position of the mobile objects as seen by the available sensors.

The SMDFM modules updates the system tracks with a rate of 1 second. At each update cycle, the SMDFM reads the mobile objects ground truth. Each sensor in the scenario has an associated software module which implements its mathematical error model and produces its perceived truth. The plots coming from the sensors are then filtered and fused in order to produce the system tracks. Figure 10 reports the sequence of operations performed at each scan by the SMDFM module.

```
// Track association
enum ObjectType { NotAssociated, Aircraft, Vehicle ];

// Track information
typedef struct TrackInfo_tag
{
    ObjectType trackType;
    unsigned long trackNum;
    octet trackStatus;
    unsigned long time;
    double x, y, altitude, speed, heading;
    string WVCat;
    boolean departing;
    string callsign,type,RWY_stand, CTD_ETA;
    string routing;
} TrackInfo;

typedef sequence<TrackInfo> TracksList;

interface TRACKS
{
    TracksList getTracksList();
};
```

5 Conclusions

A simulation environment has been developed, which allows to consider and integrate different sensor models within a complete A-SMGCS. Experimental results, obtained by considering traditional ASDE radar and GPS error models and data fusion algorithms, prove the efficiency of the simulation environment. The simulation environment is flexible and modular since it provides an infrastructure which easily allows the introduction of new software modules corresponding to different sensor models. The simulator architecture, implemented according to the Common Object Request Broker Architecture (CORBA), guarantees that new sensor error models can be easily integrated to improve the A-SMGCS simulation realism.

References