

## A Prototype Intercontinental ADS-B System Model Using MLDesigner™ and SatLab™

This application note describes a joint MLDesigner-SatLab system-level Automatic Dependant Surveillance - Broadcast (ADS-B) model. ADS-B is a proposed new Air Traffic Management System where aircraft continuously (every minute over land, less frequently over water) broadcast their GPS positions. The proposed system has planes transmit to ground stations, this model expands the concept by having the planes transmit positions via satellite, providing the ability for the position data to be sent to other aircraft as well as to ground stations. The model also incorporates an automatic conflict identification system to warn when aircraft come within a pre-defined proximity limit.) The model can easily be expanded to represent selected functions in greater detail.

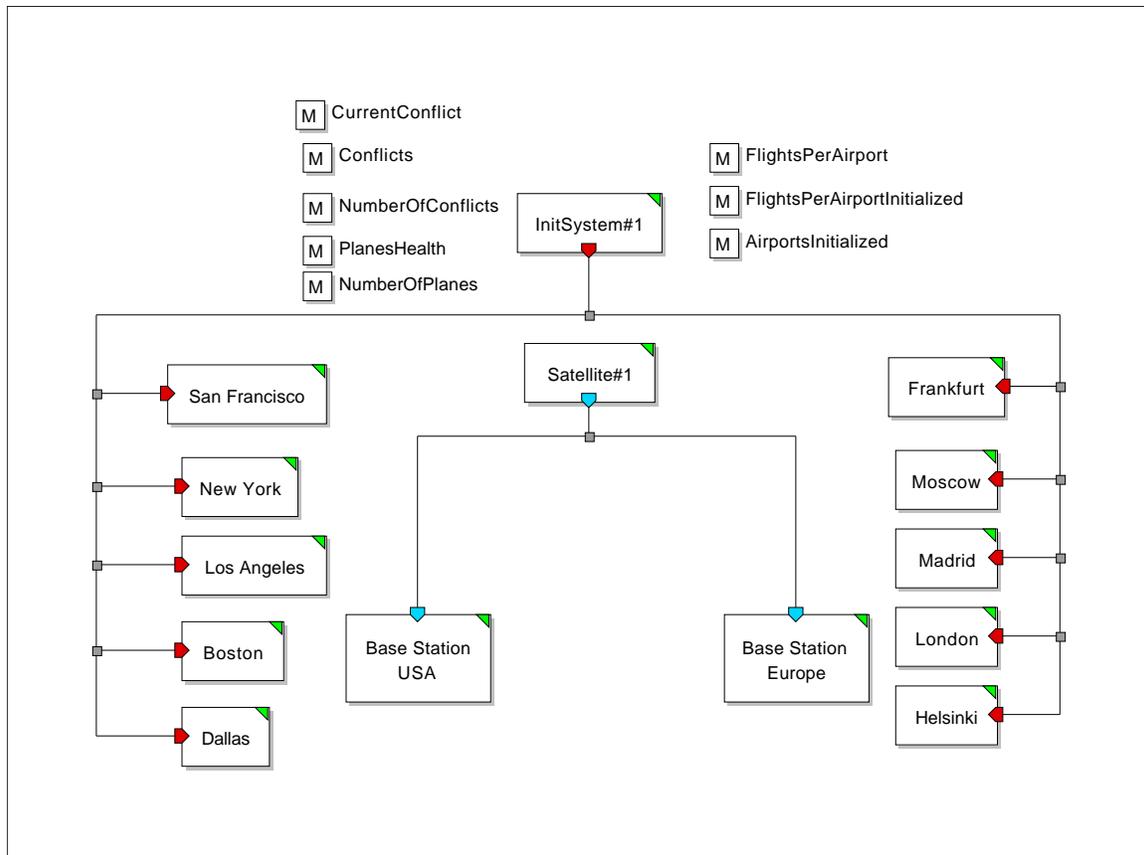
The prototype ADS-B system that uses two Air Traffic Control Centers—one in the US and one in Europe—and supports traffic between 10 airports. There are five US airports (San Francisco, Los Angeles, Dallas, New York, Boston) and five European airports (Madrid, Frankfurt, London, Helsinki, Moscow.)

The model is comprised of MLDesigner hierarchical modules representing the airports, the base stations, and the satellite system. Data structures are used to represent aircraft and flights. SatLab is linked to MLDesigner to provide position information for the aircraft-in-flight and to generate a position map showing the airports, the base stations and the in-flight aircraft.

The top-level of the MLDesigner model is shown below. It consists of 14 modules (10 airport, 2 base stations, a satellite, and an initialization module), the top-level diagram also shows the shared memories used to store information for use by all modules. The fields of the data structures used to represent aircraft in flight. The satellite module represents the operation of the communications system. It obtains and broadcasts aircraft position information (aircraft ID, time, latitude, longitude and altitude.)

The ADS-B model is parameterized so it can be quickly reconfigured. Model parameters include:

- How long the simulation should run
- Minimum/maximum flights per airport
- Number of airports
- Minimum distance to separate aircraft (A conflict will be generated when aircraft approach closer than the minimum horizontal and vertical distances.)



**Figure 1: ADS-B top level module**

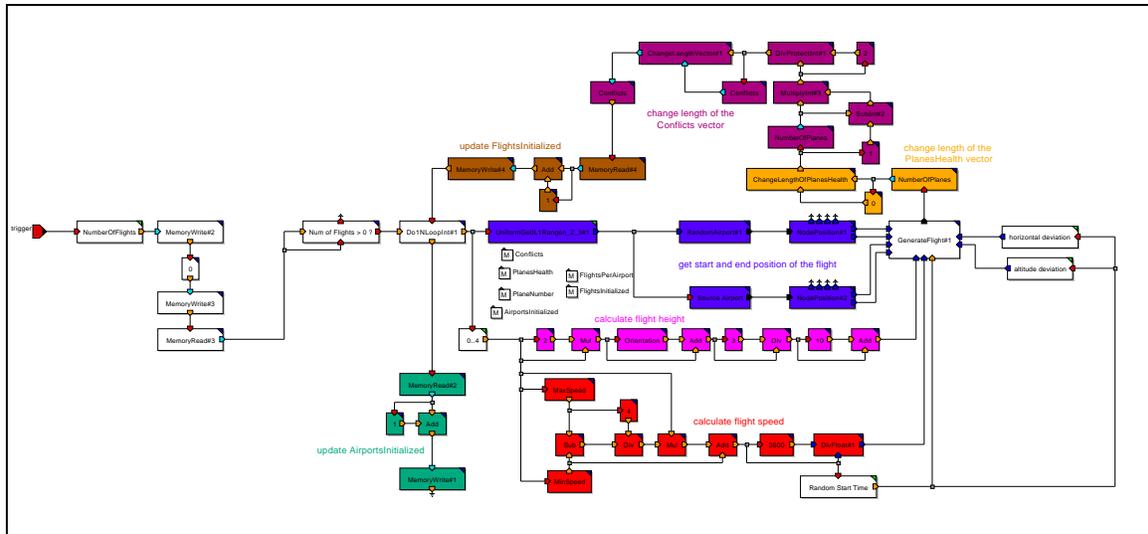
The airport module is shown below. The Airport modules generate flights and routes to destination airports and continuously monitor aircraft positions during the flight.

Most of the blocks in this module are primitives, but some, such as the three white blocks to the right center, are modules. Note that the blocks are organized by function, so the layout itself becomes a form of self-documentation.

Airport module parameters include:

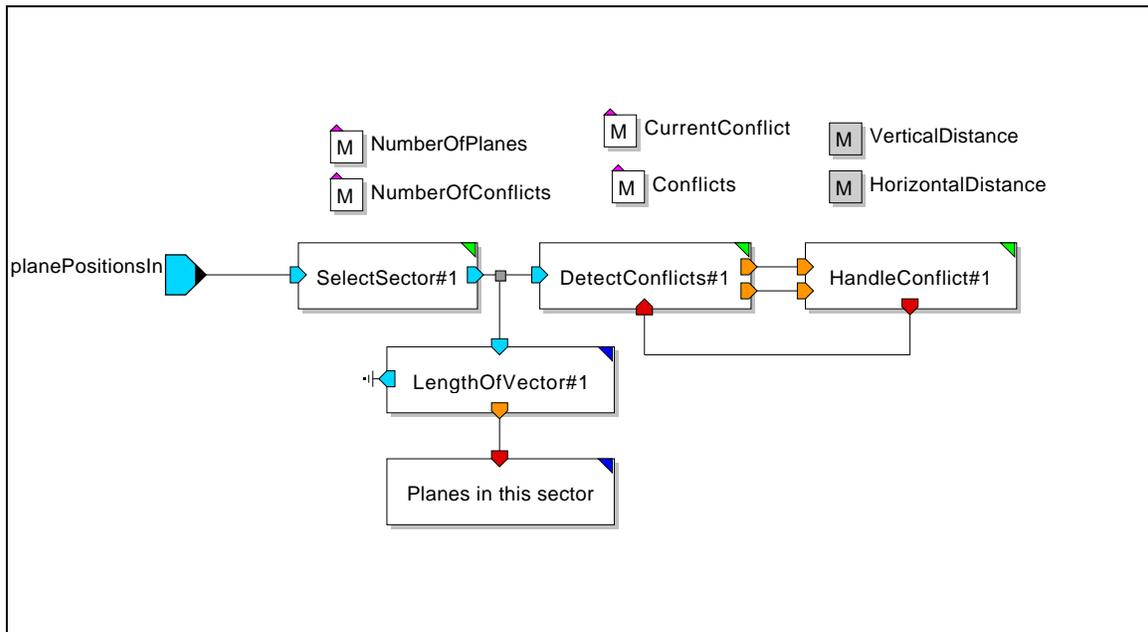
- Airport name
- Target airports
- Minimum/Maximum number of flights
- Minimum/Maximum aircraft speed
- Flight orientations (to Europe, to US)

There are ten airport modules in the current ADS-B model. (More airports could be added by adding and configuring more airport modules and by updating the parameter that sets the number of airports.)



**Figure 2: Airport module**

The ADS-B system model uses two Air Traffic Control Center (ATCC) modules to track flights, test for route/position conflicts, and resolve and report any identified conflicts. Parameters define the operational area for each base station as well as the vertical and horizontal distances required to avoid conflict warnings.



**Figure 3: Control center module**

The satellite module abstracts the position information transmission process so data structures can be used to represent flights. To do this, it calls on SatLab to provide flight position information. SatLab calculates the aircraft positions based on “flight plans” generated by the airport modules and passes the position information (altitude, latitude, longitude) to the satellite module, which, in turn, inserts the updated position information in

the respective flight data structures. (In a real system, the aircraft would obtain this information from a GPS satellite and transmit it.)

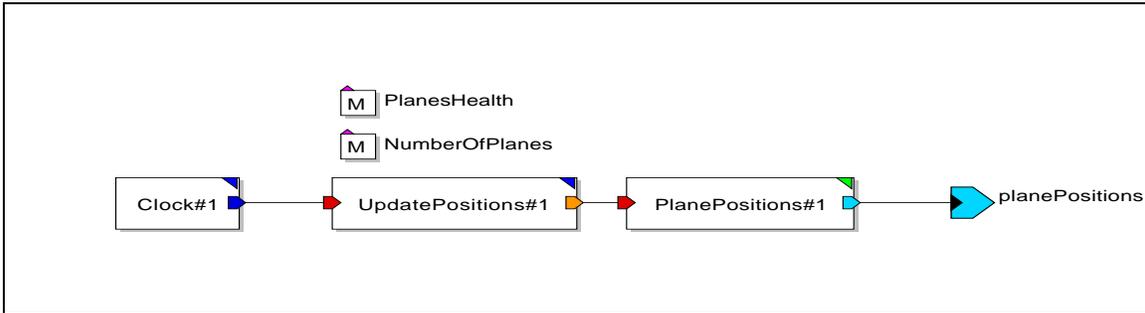


Figure 4: Satellite module

The satellite module includes modules that detect potential conflicts (where flight paths come too close) and resolve them.

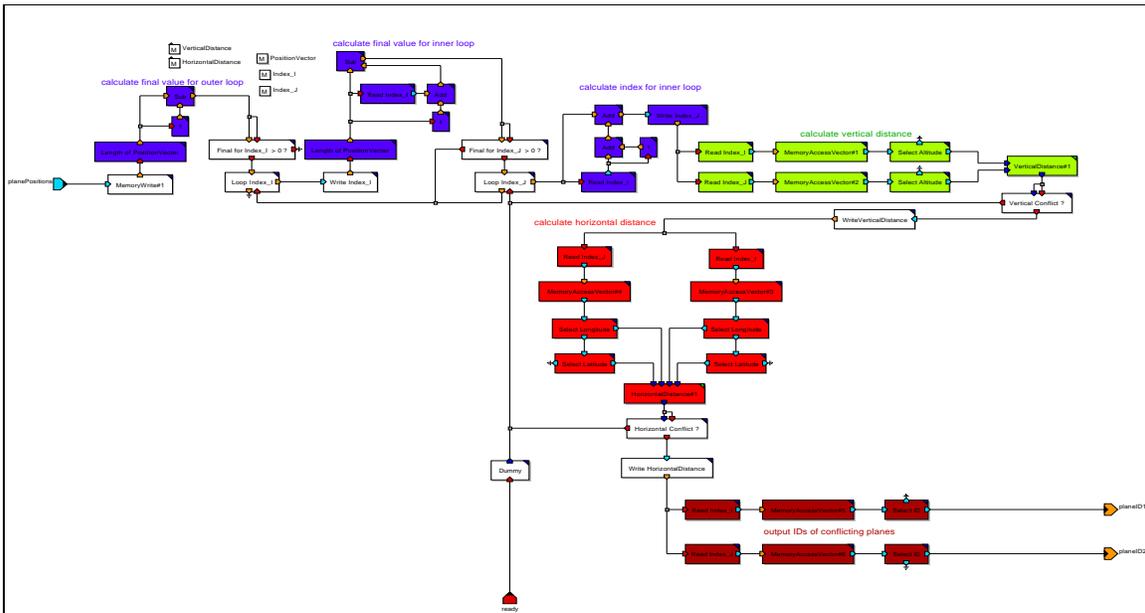
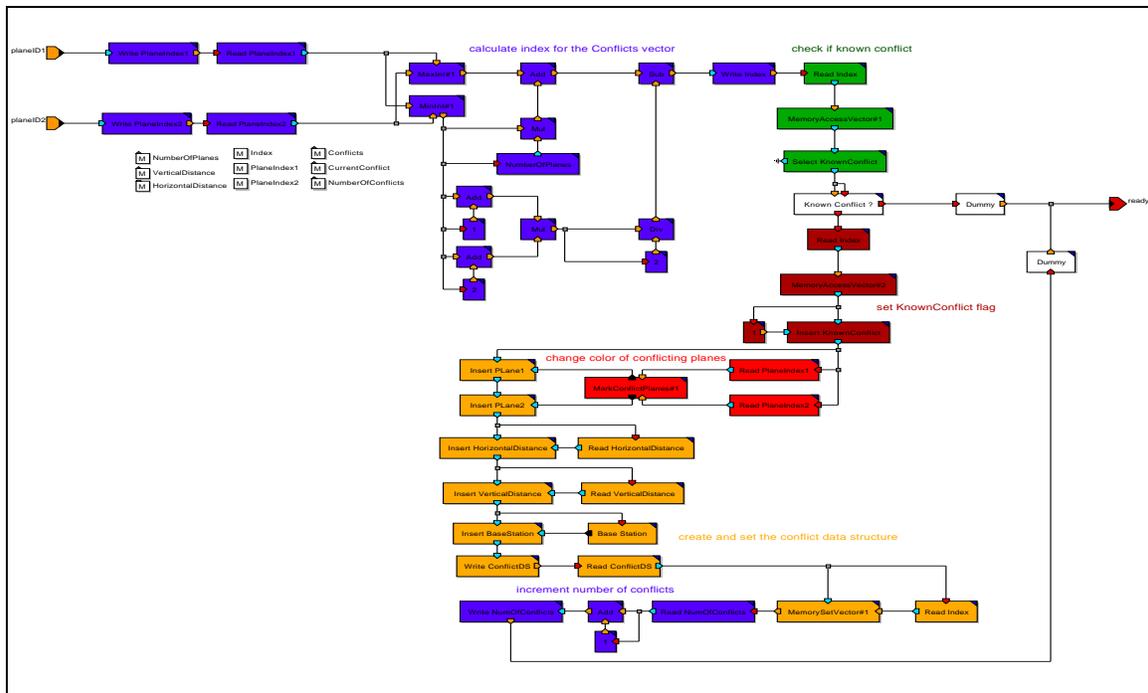
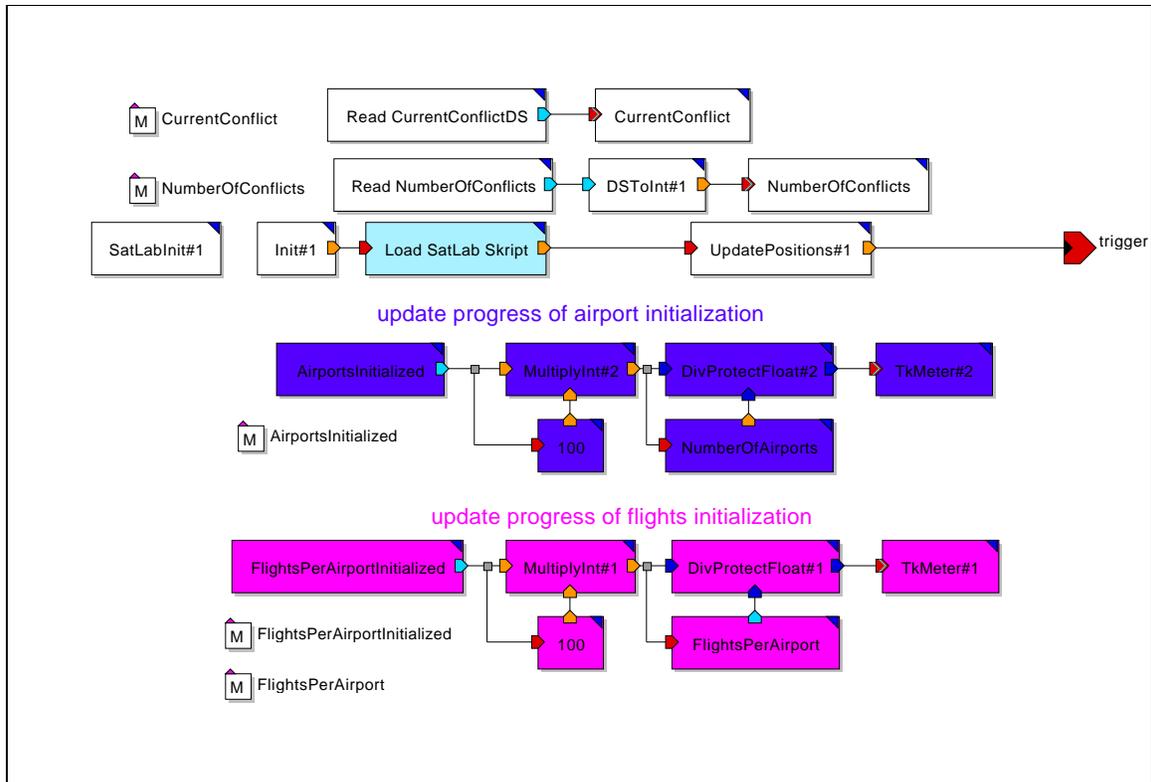


Figure 5: Detect conflicts (zoom to see detail)



**Figure 6: Handle conflicts (zoom to see detail)**

The initialization module opens communication with SatLab and configures SatLab with the appropriate script that defines SatLab operations. It also initializes the simulation model values and then starts the simulation. The initialization module opens communication with SatLab and configures SatLab with the appropriate script that defines SatLab operations. It also initializes the simulation model values and then starts the simulation.



**Figure 7: Initialization module**

Figure 3 shows the visible displays during the simulation. Typically these would be five independent displays; they are grouped here to simplify the presentation. The SatLab map to the left shows the locations of the airports and the base stations. It also shows the positions of all active flights (shown by yellow dotted lines.) The top window to the right is used to display progress during initialization.

The next two windows show fight-in-progress reports for the US and European base stations.

The bottom two windows show a combined (US and European) conflict report and the details of each conflict. (Conflicts are triggered whenever aircraft approach closer than the specified minimum distance.)

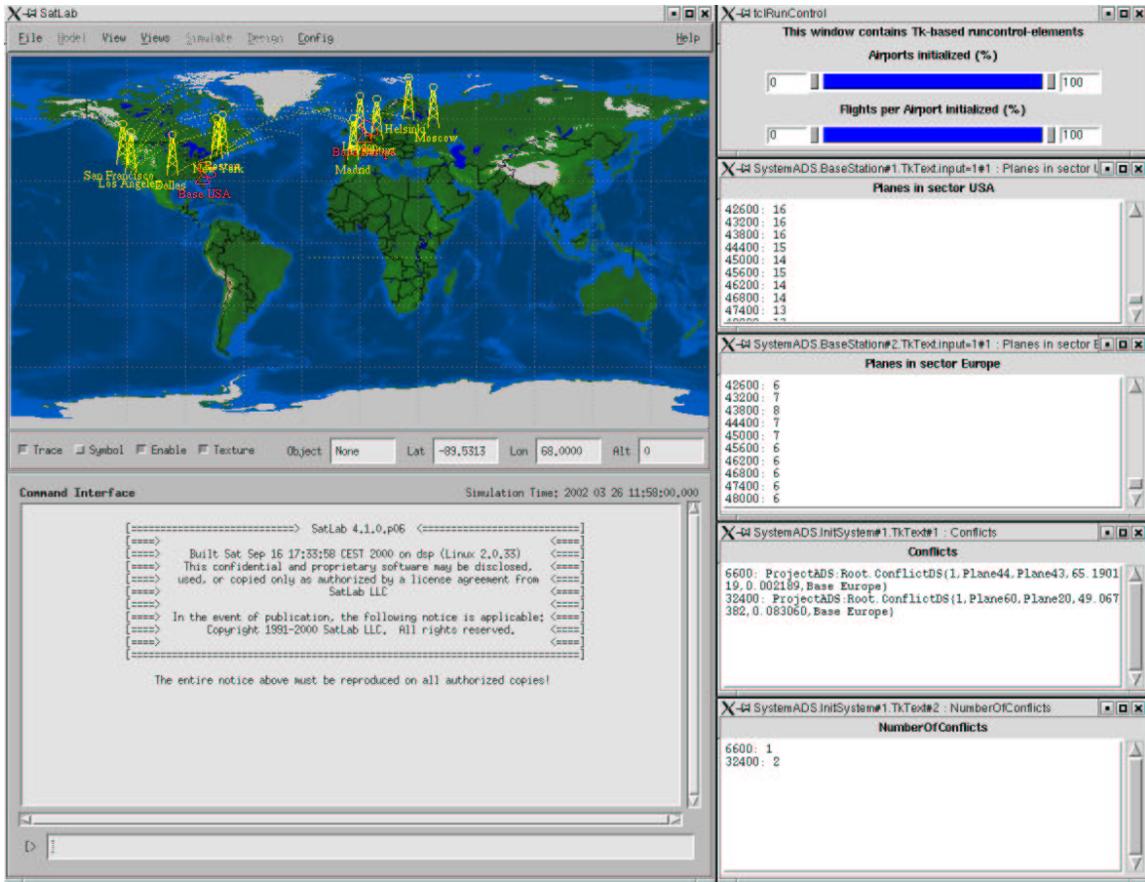


Figure 8: SatLab map plus MLDesigner report windows

*Note: A 10-page white paper that provides more details about this model is available.*