## Document Change Log

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Change Description</th>
<th>Author</th>
<th>Reviewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/06/2011</td>
<td>Initial</td>
<td></td>
<td>Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>11/9/2011</td>
<td>2nd</td>
<td>Added descriptions for new features of violation</td>
<td>Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>11/23/2011</td>
<td>3rd</td>
<td>Added description of data validation tool and FCFS</td>
<td>Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>12/27/2011</td>
<td>4th</td>
<td>Added Post Analysis Description</td>
<td>Sergei Gridnev</td>
<td></td>
</tr>
<tr>
<td>3/9/2012</td>
<td>5th</td>
<td>Add new CD&amp;R</td>
<td>Zhifan Zhu</td>
<td>Sergei Gridnev</td>
</tr>
<tr>
<td>8/8/2012</td>
<td>6th</td>
<td>Updated figures</td>
<td>Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>11/15/2012</td>
<td>7th</td>
<td>Updated CD&amp;R</td>
<td>Sergei Gridnev</td>
<td>Zhifan Zhu</td>
</tr>
<tr>
<td>12/28/2012</td>
<td>8th</td>
<td>Update CD&amp;R, added Uncertainty section</td>
<td>Sergei Gridnev, Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>06/12/2013</td>
<td>9th</td>
<td>Added Playback</td>
<td>Sergei Gridnev</td>
<td>Zhifan Zhu</td>
</tr>
<tr>
<td>09/30/2013</td>
<td>10th</td>
<td>Deleted Post Analysis and updated new gui menus</td>
<td>Zhifan Zhu</td>
<td></td>
</tr>
<tr>
<td>12/03/2013</td>
<td>11th</td>
<td>Updated CDR and Alerts controls, View and preferences menu options, Removed GnuPlot dependencies</td>
<td>Sergei Gridnev</td>
<td></td>
</tr>
<tr>
<td>3/14/2014</td>
<td></td>
<td>Updated CAI 1.9 support</td>
<td>Sergei Gridnev</td>
<td></td>
</tr>
<tr>
<td>11/10/2014</td>
<td></td>
<td>Updated/Added Aircraft List, Dynamic Scenario Input, Simulation Configuration, Uncertainty Configuration, Critical Area Control, Dynamic Scenario Input</td>
<td>Sergei Gridnev</td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

1 INTRODUCTION ........................................................................................................... 4
   1.1 PLATFORM OS ......................................................................................................... 4
   1.2 SOFTWARE DEPENDENCY ..................................................................................... 4
   1.3 BUILD ..................................................................................................................... 4
       1.3.1 The software provides build scripts using cmake as following steps.............. 5
   1.4 RUN ......................................................................................................................... 5
   1.5 DATA DIRECTORY .................................................................................................. 5
       1.5.1 Airport Adaptations ....................................................................................... 6
       1.5.2 Traffic Scenario Data ...................................................................................... 6
       1.5.3 Aircraft Types .................................................................................................. 7

2 SOSS GUI .................................................................................................................... 7
   2.1 MAIN WINDOW ...................................................................................................... 7
       2.1.1 File .................................................................................................................. 8
       2.1.2 Edit .................................................................................................................. 8
       2.1.3 View ............................................................................................................... 8
       2.1.4 Preferences .................................................................................................... 9
       2.1.5 Tools ............................................................................................................... 9
       2.1.6 Help ............................................................................................................... 10
       2.1.7 Mode .............................................................................................................. 10
       2.1.8 Status ............................................................................................................ 10
   2.2 AIRPORT PREVIEW ............................................................................................... 11
   2.3 SCENARIO FILE ..................................................................................................... 11
   2.4 CONFIGURATION FILE ......................................................................................... 11
   2.5 CONFIGURE SCHEDULER .................................................................................... 11
   2.6 SIMULATION CONTROLS ...................................................................................... 12
   2.7 TRAFFIC MANAGEMENT INTERFACE CONTROL ........................................... 15
       2.7.1 Estimated Departure Clearance Time (EDCT) .............................................. 15
       2.7.2 Departure Fix Separation Control (MIT) ..................................................... 15
   2.8 UNCERTAINTY CONTROL ..................................................................................... 16
       2.8.1 Speed Uncertainty ......................................................................................... 18
       2.8.2 Time Uncertainty .......................................................................................... 18
       2.8.3 Constant Speed Modifiers ............................................................................ 19
   2.9 RUN SIMULATION ................................................................................................. 19
   2.10 AIRCRAFT LIST .................................................................................................... 19
   2.11 VIEW CHECK BOXES .......................................................................................... 21
   2.12 SAFETY VIOLATION ALERT DISPLAY ............................................................... 25
   2.13 MOUSE CONTROLS ............................................................................................. 26
   2.14 TOOLS .................................................................................................................. 26
1 Introduction

This User Guide describes SOSS software build and graphic user interface.

SOSS is a desktop application that simulates airport surface operations in fast time using traffic management algorithms. It moves aircraft on the airport surface based on information provided by scheduling algorithm prototypes, monitors separation violation and scheduling conformance, and produces scheduling algorithm performance data.

The diagram in Figure 1 shows the SOSS operation environment.

1.1 Platform OS

SOSS has been developed in cross-platform code. But at this release time, it has been built and tested only on Linux and Mac platforms.

1.2 Software Dependency

Two open-source software packages are required to build and run SOSS. They must be installed before building SOSS:

- Boost c++ library [www.boost.org](http://www.boost.org) (version 1.47 or 1.48)
- Qt 4 library (version 4.6 and above)

1.3 Build
1.3.1 The software provides build scripts using cmake as following steps

- Copy the source code trunk to your system (e.g., `/your_home/soss_source`)
- Create a directory where you want to install the software, e.g., `/your_home/soss`.
- In the installation directory, type `cmake /your_home/soss_source`
- If cmake goes through successfully, type `make` to build

Note: the BOOST_ROOT variable at top level CMakeLists.txt is set default as `/usr/local/boost` to reflect the Boost installation at our development environment. It need be changed if your Boost is not installed at `/usr/local/boost`. Setting this variable is due to Boost’s the known Boost cmake script issue.

Successful build will create the binaries in bin directory:

- libSOSSAlgorithms – some internal algorithms that SOSS uses
- libSOSSEngine – the main SOSS engine
- soss – a command-line application that uses SOSS engine
- sossGUI – a GUI application to run SOSS engine
- dataValidate – a command-line tool to check soss input data

1.4 Run

SOSS executable can be launched from the bin directory from your installation location. See the GUI user guide section for sossGUI and command-line section for soss.

1.5 Data Directory

The data directory consists of airport adaptations, scenario data, and aircraft type database files that SOSS need. User can add additional airport adaptations and traffic scenarios late. The diagram in Figure 2 shows an example of data structure with two airports, DFW and JFK.
1.5.1 Airport Adaptations

The airport models are under each sub-directory of its name, e.g., DFW and JFK. The airport model data include:

- nodes.txt – node data
- links.txt – link data
- fixes.txt – fix data
- runways.txt – runway data
- runwayConfigurations.txt – runway configuration data
- separations.txt – separation rules
- routes.txt – a lookup-table like static taxi route data
- airport.txt – airport data
- CriticalArea.txt – defines exclusive areas for another level of ac-to-ac separation on runways (optional)

Separation rules consist of matrices for different runway operations. Rules are airport configuration dependent.

1.5.2 Traffic Scenario Data

Traffic data are given in scenario data files under data/scenario_data. User should not make changes to the file contents. New scenario data files can be brought in when needed.
Note that scenario data file embeds airport name and runway configuration information. User should not change the embedded information in scenario data file.

1.5.3 Aircraft Types

There are two data files under this directory, named `aircraft_types_data` and `initial_aircraft_model_properties`, respectively. They provide aircraft characteristics based on aircraft types. SOSS uses the information to model aircraft mobility. They should not be changed.

2 SOSS GUI

SOSS GUI allows a user to run SOSS interactively. It provides graphic interface controls and visualization of SOSS simulation run.

2.1 Main Window

To launch SOSS GUI application, type command `./sossGUI` in bin directory. It will bring up the SOSS GUI window shown in Figure 3.

![SOSS GUI Window](image)

**Figure 3**

The SOSS GUI window consists of a main function menu bar, a short cut tool bar, a simulation status progress bar at the bottom, and an airport map display area in the center. Depending on the mode selected the tool bar options and the progress bar will vary in functions and appearance.

The main function menu bar includes seven menus.
2.1.1 File
The commands in this menu allow the user to choose an airport to preview or choose a scenario to run a simulation. In addition there are commands to save the SOSS configuration, post simulation data, and database to a file.

![File Menu](image)

Figure 4

2.1.2 Edit
Commands in this menu allow the user to configure scheduler, runtime parameters, uncertainty generators, and traffic management control parameters that determine how a SOSS simulation will be run.

![Edit Menu](image)

Figure 5

2.1.3 View
The View menu contains commands to show aircraft list and toggles to turn on and off some display options.
2.1.4 Preferences

The Preferences menu allows user to change colors in the map display.

2.1.5 Tools

The Tools menu groups the functionalities for details of the simulation. Currently, Show Delays and Show Scheduler Call History tools have not been implemented yet.

Safety violations shown only contain separation violation between two AC on taxiway.
2.1.6 Help

The Help menu has two options: 1) search SOSS help topics., and 2) show the color legend of nodes and links.

2.1.7 Mode

The Mode menu allows the user to select and check the current SOSS mode (Simulation or Playback).

2.1.8 Status

At the bottom of the main window is the status bar. It consists of the following information:

- Airport and scenario data file
- Simulation progress in percentage of aircraft completion
2.2 Airport Preview
Choose File | Load Airport and choose one airport to preview its model.

2.3 Scenario File
Choose a scenario file from File | Load Scenario to load the traffic scenario. It will load corresponding airport model.

Traffic scenario data is the primary input data for simulation. This is the first step to start to simulate a new traffic scenario. The general procedure to run a SOSS simulation without configuration data file with GUI is:

- Select scenario data
- Configure scheduler and simulation parameters
- Start run simulation
- Analyze results

User can save the configuration parameters to a configuration file to avoid setting up those parameters repeatedly.

Once a scenario data is loaded, SOSS provides a default set of configuration parameters. After user’s modification, the parameters can be saved by using File | Save Configuration as command.

2.4 Configuration File
Configuration file allow loading previously saved configuration to SOSS. Since scenario data file is part of the configuration, loading through a configuration data file will load the scenario data and airport at the same time.

- Choose File | Load Configuration and select a previously saved configuration file from the file dialogue.
- Choose File | Save Configuration or click on the short cut Save button in the tool bar to save configuration changes..
- Choose File | Save Configuration as or clicking on the short cut save as button in the tool bar to save configuration changes to a new file.

2.5 Configure Scheduler
Choose Edit | Scheduler or its tool button to open a dialog window to configure scheduler(s).
To provide interaction with various versions of Schedulers CAI Version drop box allows selection of the communication format that will be used for this particular run.

A scheduler configuration consists of following parameters:

- Scheduler Name – a name for the scheduler
- Scheduler address – this is the IP address and port number
- Call Intervals in seconds – this defines how often the scheduler is called
- Forecast Window in minutes – this is the window size that SOSS use to send aircraft information to scheduler for scheduling. E.g., a 10 minutes window will send those aircraft that is on surface and will be on surface within the window to scheduler
- Initial Call – this is the time that the scheduler will be called first time

At the bottom of the dialog shows the schedulers. User can add a new scheduler and remove a scheduler.

### 2.6 Simulation Controls

Choose Edit | Simulation or its tool button to open the dialog for simulation parameters.
Figure 12

- Aircraft Guidance Mode – Open Loop or Close Loop
  - In Open Loop mode, SOSS moves aircraft using a command speed based on its nominal speeds.
  - In Close Loop mode, SOSS moves aircraft using a command speed within a lower and upper bound about its nominal speed. The upper bound speed is 120% of the nominal speed, and the lower bound is controlled by a Minimum Speed Factor (default is 0.75).
- Minimum Speed Factor – see above Close Loop Mode
- Time Step in seconds – this is the simulation delta time. Default is 0.5 seconds.
• Fast Time Gain – this is the desired fast time gain to control the speed of the fast time simulation. E.g., a 20 gain will finish simulation of 3600 seconds in about 180 clock seconds (per hardware).

• Time Pad after Last PTime – this is the amount of time in minutes added to the last PTime of all aircraft such that simulation will be forced to stop. In normal situations, simulation will stop when all aircraft have reached their surface destination. This is to prevent simulation from running forever in case of gridlock. Default is 30 minutes.

• Arrival Gate Occupancy Duration – the amount of time, in minutes, that an arrival aircraft will remain on the surface after reaching the gate.

• Departure Gate Occupancy Duration – the amount of time, in minutes, that a departure aircraft will be at gate before scheduled push back (PTime).

• Wind Direction – airport wind direction in true north, in degrees (“winds from”) – used by landing and take off models

• Wind Speed – airport wind speed, in knots – used by landing and take off models

• A matrix of checkboxes is available to enable Safety Alerts and Conflict Detection and Resolution (CDR) in various airport surface areas. Rows represent specific surface areas while columns represent Safety Alerts and CDR functionality and are as follows:
  
  o Ramp Area – physical separation violation between gate and spot.
  o Movement Area – physical separation violation between spot and runway.
  o Runways – violation of runway separation rules at runway crossings and departure nodes.

• CDR checkbox controls enable violation resolution in three airport areas
  
  o Ramp Area – conflict detection and resolution using FCFS between gate and spot.
  o Movement Area – conflict detection and resolution using FCFS between spot and runway.
  o Exclusion Area – conflict resolution using FCFS and arrival-on-landing priority that allows only one aircraft in a user defined geometric area
  o Runways – enforcement of user-specified runway separation rules for aircraft taking off or crossing a runway.

• Prediction Window Size is the look-ahead window in seconds used in safety violation prediction.
• Prediction Delta Factor is the multiple of simulation time steps that the prediction algorithm uses to predict aircraft positions in equal times. E.g., a value of 2 for 0.5 second simulation time steps makes the prediction delta time one second.

• Long Separation FactorA is the distance in meters that defines the minimal separation threshold requirement between aircraft. For instance a value of 0 would indicate the user preference to allow the nose of one aircraft to almost touch the tail of the other. The recommended setting is 10 to avoid “fender benders” in certain queue situations.

• Long Separation FactorB controls a dynamic distance buffer that uses a product of itself and the speed of an aircraft for additional separation requirement in head-to-tail situations. Thus when an aircraft A tailgates B the required separation threshold between them used by CD&R will be: Long Separation FactorA + max(speed of A, speed of B) x Long Separation FactorB. This is an important setting to prevent “pile on” violations when a leading aircraft is forced to suddenly stop (especially when its model allows for faster deceleration than the aircraft in the back). The recommended setting is 5.

• Block Scheduler Calls – if checked, SOSS will call scheduler(s) in blocking mode.

• Runway Sequence Control – it allows selection of runways to enforce aircraft runway access (departure and cross) sequence derived from runway usage times give by scheduler(s).

2.7 Traffic Management Interface Control

SOSS design has two types of traffic management control configuration. Choose menu **Configuration | TMI | EDCT** to set the Expected Delay Clearance Time for departure aircraft. Choose **Configuration | TMI | Dept Fix Range** for Departure Fix Separation Control.

2.7.1 Estimated Departure Clearance Time (EDCT)

Not implemented yet.

2.7.2 Departure Fix Separation Control (MIT)

Choose **Edit | TMI | Departure Fix MIT** to open the dialog:
The dialog shows a table of all departure fix and their MIT in seconds. User can change the MIT values.

Click **OK** button to accept the configuration, or **Cancel** to cancel it.

### 2.8 Uncertainty Control

Figure 14 shows the configuration dialog window that user can configure uncertainties during the simulation.
There are two types of uncertainties. One is aircraft speed uncertainty that affects aircraft nominal speed in each movement area. The other is delay uncertainty affects aircraft push back process.

### 2.8.1 Speed Uncertainty

There are three speed uncertainties for ramp, taxi and queue areas, respectively. The uncertainty is represented by a random variable value to be added to aircraft’s nominal speed. For instance, if an aircraft’s taxi area nominal speed is 14 knots, Operation Rule is set at Once, and a random number of 2 knots is generated for the taxi area uncertainty, then the aircraft will move at a target speed of 16 knots in taxi area.

Each uncertainty has the following configurable selections.

- **Add uncertainty** – to switch on or off the uncertainty (default is No)
- **Distribution type** – Uniform, Normal, Gamma, and Cauchy
- **Operation rule** – how often the random variable gets generated/re-generated for each aircraft
  - Once Only – SOSS generates a single uncertainty random value for an aircraft at the beginning of a simulation. The target speed of the aircraft will stay the same during the simulation.
  - Every Simulation Iteration – SOSS generates a new uncertainty random value for an aircraft every simulation time step. The target speed of the aircraft will change quickest during simulation.
  - Every Several Seconds – SOSS generates a new uncertainty random value for an aircraft every constant period of time, set by the user. The target speed of the aircraft will change regularly during the simulation.
  - Every time AC accelerates – SOSS generates a new uncertainty random value when an aircraft starts to accelerate from stop.
- **Standard deviation** – standard deviation of distribution
- **Seed value** – user selected random sequence seed value

The default mean values of the speed random variable are all zero.

There is a global lower bound and upper bound that user can set on the configuration with the minimum and maximum speed limits. They are global settings and apply to all aircraft and all surface areas. They effectively truncate the uncertainty distribution within the bounds. In the previous example, if a maximum 15 knots speed limit is set, then the target speed will be truncated to 15 knots rather than 16 knots.

### 2.8.2 Time Uncertainty

There are two time uncertainties. They are related to departure aircraft push back process only.

The first one is called Flight Readiness Uncertainty. It adds an extra random delay for an aircraft to start push back from the gate. For instance, if an aircraft is scheduled to start push back at PTime =
100, and Flight Readiness Uncertainty causes an extra 10 second delay, then the aircraft will start push back (leaving the gate node) at time = 110 during simulation. Note that a scheduler may request departure gate holding. Using the previous example, if a scheduler requests the aircraft hold at the gate until time = 120, then the aircraft will start push back at that time; if scheduler’s gate holding time = 105, the aircraft will start push back at time = 110. In other words, the aircraft will its gate at maximum of PTime + Flight Readiness Uncertainty and scheduler holding time (STR).

The second one is Push Back Uncertainty. The push back process is defined on the first link (from gate). The effect of this uncertainty is on the push back speed and ultimately the time the aircraft arriving at the end of the first link. In this case, a nominal push back time is calculated from the aircraft’s nominal push back speed and the length of the link. Then a random amount of time from the Push Back Uncertainty is added to the nominal push back time. Finally, the effective push back speed is calculated as the link length divided by the total time. As a result, the aircraft will arrive at the end of the link with the uncertainty delay.

Like speed uncertainties, these two delay uncertainties parameters can be configured at the dialog window. The operation rule for the two uncertainties is always Once Only.

2.8.3 Constant Speed Modifiers
These controls allow the user to adjust nominal aircraft speed for pushback, Taxi, Ramp, and Queue areas. These are constant, rather than random, values and will be added to the existing nominal speed values.

2.9 Run Simulation
Once scenario data is loaded and configuration is done, simulation can be started, paused, and stopped by clicking the Start, Pause, and Stop tool buttons at the tool bar.

2.10 Aircraft List
Choose View | Aircraft List to view the aircraft list in a pop up dialog. Each aircraft shows its CallSign, Aircraft Type, Category, Spot Name, Runway, Gate Name. The font color in all columns but “P Time” corresponds to the values selected by user for arrival and departure aircraft icons from “Preferences”. The background color of all cells is the same value as selected by user for airport background. The P Time column font is green if the aircraft is active, and red otherwise. Initially, the dialog is shown in a float window.
Figure 15

User can drag and drop the window to the main display area to dock the dialog:
2.11 View Check Boxes

From View menu, user can select and deselect viewing options.

*Choose View | Show Node Link* to toggle display of airport node-link model. This option is on by default.
Choose View | Show Airport Map to toggle display of airport background map.
Choose View | Aircraft Callsign to turn on/off aircraft information block display in the map. The block currently has three numbers: aircraft call sign, aircraft type, and current speed in knots.

Choose View | Overlay to toggle display some runtime information on the map.
Choose View | Node Label and/or View | Link Label toggles display of node and/or link labels. User needs to zoom in to see the labels. SOSS only display node and link labels in details to avoid map being crowded.
Choose View | Taxi Route to display the taxi route for the selected aircraft (from the Aircraft List). The taxi route is displayed for one aircraft only. User need select an aircraft from the Aircraft List.

A taxi route displays the route and three times at a node if it has received scheduled time. The three times are: scheduled time release (leaving from the node), actual time arrival, and actual time release.

Choose View | Show CDR Dependency to display the action CDR is taking to resolve conflicts between aircraft. An arrow will originate from a yielding aircraft and point to one that has right of way. Red color arrow denotes head-on conflict resolution while yellow stands for tail-on conflicts.

2.12 Safety Violation Alert Display

When Safety Violation Alert is on, SOSS will show aircraft with violation with a distinct color and a flashing circle about the aircraft.
2.13 Mouse Controls

Current mouse controls allow pan and zoom in the airport display.

- Drag while holding right mouse pans the map
- Wheel up and down zooms the map

2.14 Tools

2.14.1 Plot AC Speed

After simulation has finished, this tool allows plot selected aircraft speed against time. User can use left mouse to create a rubber band on the graph to zoom in and click the zoom level button to zoom out.
2.14.2 Show Aircraft to Aircraft Safety Violations

After simulation has finished, this tool displays safety violation alerts between two aircraft on surface between terminals and runways.

The violations are sorted with simulation time. Each row shows a safety separation violation between two aircraft that happened at first time. The aircraft call signs are shown in the columns following the time, followed by the required separation and actual separation in meters.
Runway safety separation violations are between two aircraft when accessing runways. Each occurrence is recorded at the time the first happens. Runway separation is measured in time (of seconds). A runway separation rule requires safety separation in a same runway or a coupled runway operation. The runway shown in the display is the runway the second aircraft (AC2) access that caused a runway separation violation due to the first aircraft (AC1) runway operation (the same or other runway). The required and actual separation columns are in seconds. The detail of the violation is given in the last column.
### 2.14.4 Show Delays

TBD

### 2.14.5 Show AC Route Times

This tool displays, for a selected aircraft, the times along taxi route. A ‘-1’ is used in place that no data is available. Furthermore, user can highlight any one particular table row, which will highlight the corresponding node on airport surface with a yellow square.

![Runway Separation Violations Table](image-url)
2.14.6 Show Scheduler Call History

TBD

2.14.7 Show Runway Usage History

Runway Usage History shows runway use times sorted by runway and then times. Two runway uses are included: runway departure and runway crossing, distinguished by Use Type column.

Runway departure rows have departure Fix ID and runway-crossing rows have crossing name.
## 2.14.8 Show Departure Fix History

The departure fix usage table shows the departure fix being used by aircraft information. It has five columns:

- Departure Fix (name)
- Departure Fix (Id)
- Time (of departure fix) – this is the time a departure aircraft starting taking-off
- Call Sign (of the aircraft)
- Weight Class (of the aircraft)
2.14.9 Zoom to Node/Link

This tool allows user to zoom to a node or link. Only node/link index is supported.

2.15 CDR Dependency

Choose View | Show CDR Dependency to view CDR interaction between conflicting aircraft. If CDR is enabled and active this tool will show the resolution implemented by the CDR.
module for every conflicting pair. An arrow pointing from a yielding aircraft to the one with right of way visualizes the resolution. The color of the arrow is a key to conflict type, where red denotes head-on conflict while yellow corresponds to tail-on.

### 2.16 Playback Mode

Playback mode allows user to replay and closely examine a previously ran scenario through a variety of features.

#### 2.16.1 Switching Mode

Switching between Simulation and Playback modes is achieved through a drop down "Mode" menu one can find among the top menu bar items of SOSS GUI. Switching modes can only be achieved when SOSS is not running in either mode. In other words you have to stop playback or simulation in order to enter a different mode.

#### 2.16.2 Playback Overview

The playback database may contain only partial simulation data, depending on how far the original simulation has ran its course. Thus even a partial 10 second sim run can be stored and/or replayed in this mode.

The user may change the direction or speed without pausing or stopping the playback.

A freshly loaded playback will start at $t = 0$ in forward play or at $t=\text{scenario\_length}$ in reverse play.

If there is no playback data for a particular time step in a partial playback database the aircraft will either stay at their last know location or will not be displayed at all, depending on the direction of playback.

Playback configuration file that is stored along with the database points to the original scenario, aircraft, and airport data. This data must be present during playback in order for many GUI tools to work, such as displaying of aircraft routes, airport surface map, etc.

SOSS stores the its previous GUI mode along with scenario/playback data. When restarted it will returned to the same mode it was at closing time as well as load the original scenario/playback database. Note that for the second use case when a playback database was not stored it will revert to last known binary playback database.

#### 2.16.3 Playback Database

The database is an sqlite3 DB in binary format with an extension *.playdb and contains aircraft state information that allows SOSS to display AC position during playback. It currently does not contain any CDR data. Along with the database a copy of a configuration file is saved under the same root name but with a *.txt extension.

#### 2.16.4 Playback Controls

In Playback a toolbar specific to this mode will appear at the top of the main SOSS GUI window as seen in Figure 31.
Almost all simulation data are kept in an internal (in memory) database. They can be selectively saved to text files with the Save Simulation Data To File option under the File Menu. The database itself can also be dumped to an external database file for late retrieval (e.g., for playback).

2.17 Save Simulation Data

After simulation has finished, select File | Save Simulation Data to Files allows user to select a directory to save following data results. All data files are in plain text format with header information.

- ACStateHistoryData.txt – all aircraft dynamic information at each simulation time step
- ACRTACConformance.txt – scheduled and actual times for each aircraft
- ACScheduleData.txt – aircraft times at each node of taxi route
- SchedulerExecution.txt – scheduler call history
- SchedulerStability.txt – scheduler requested times for each aircraft
- SafetyViolation.txt – safety violation information
- AirportScheduleData.txt – airport runway usage information
- Configuration.txt – simulation configuration data
- GUIPreferences.txt – gui preference data
2.17.2 Save Simulation to Database
After simulation has finished, select File | Save Database or Save Database As allows user to select a directory to save the whole internal database to a file that can be loaded back at a later time. This is useful for playback.

3 Command-line SOSS
SOSS can also be run using its command-line build, soss. To run SOSS command-line application, a SOSS configuration data file is needed as the input:

```
./soss configuration_data_file.
```

There is an example configuration included under the test directory. The configuration file is in plain text format. It is advised to use SOSS GUI to configure the parameters and save it to a configuration data file.

4 Command-line Data Validation Tool
A SOSS input data check tool, dataValidate, is included to help verify the integrity of the input data to SOSS. It is a command-line executable that takes a scenario data file as the command-line argument. It is highly recommended to run this tool to check new scenario data and new airport adaptation files.

The tool has to be run in the bin directory of SOSS build to make sure it can access referenced airport and aircraft data files. For instance, to check a scenario data named MyScenario.list_data:

```
./dataValidate MyScenario.list_data
```

The dataValidate tool will:

- Parse the scenario data
- Load airport model and set runway configuration (specified in the scenario)
- Verify the loaded airport model (nodes, links, fixes, runways)
- Verify existence of static taxi routes for all flights
- Validate tail number dependency mapping for turnaround aircraft

If a SOSS exception (such as a runway node is not defined in the node data) is caught, dataValidate will stop and display an error message.

Some non-critical errors (such as a static taxi route cannot be found between a terminal and runway for a flight) are found during the process, dataValidate will display the error messages.

The tool also shows some information messages such as runway departure, arrival and crossing nodes.
5  Surface Conflict Detection and Resolution
For details of CD&R logic and data output, please refer to document ‘SOSSCDRLogicDescription.pdf’

6  Runway Sequence Control (RSC)
“Runway use” for the purpose of RSC is defined as an act of traversing across a runway crossing or as taking off from a runway. Thus “runway use node” will be the first crossing node in case of the former, and the departure node for the later. Runway Sequence Control is a user-selected option that mandates every aircraft to satisfy its requirement before proceeding with runway use. This option can be selected independently for every runway and necessitates the use of a scheduler. When enabled, the RSC algorithm will compare the Scheduled Release Time (SRT) of the runway use node received from the scheduler for every aircraft that will use a controlled runway. The aircraft call signs will be entered into a queue and sorted in ascending order based on their respective SRTs. In the case when an aircraft was not issued an SRT for a runway use node by the scheduler it will be omitted from the queue. The scheduler can change aircraft’s position in the queue by issuing a new SRT with every successive call. An aircraft is allowed to use the runway by RSC only if it occupies the front most position in the queue. For example when aircraft A tries to take off and aircraft B is trying to cross the same runway the right of way will be determined by the front position in the queue, regardless of which aircraft got to their respective runway use nodes first. Once the runway operation is completed the aircraft will be removed from the queue.

One of the reasons to enable RSC would be to prevent runway use starvation of one aircraft by others when it is at a disadvantage by imposed Runway Use Separation rules.

7  Critical Area Control
This feature is an addition to runways safety separation control. It allows the user to define a polygon by specifying the x,y coordinates of at least three vertices within CriticalArea.txt file. Thereafter if Critical Area control is enabled only one aircraft at a time will be allowed within this area at any given time. The aircraft will queued up on a first come first serve basis. This feature was designed and is particularly useful for preventing aircraft entering runway boundaries in multiple departure nodes per runway scenarios, like HAM. Using Critical Area Control an aircraft will not wait on the runway surface at one departure node while being “overrun” by another aircraft taking off from a departure node further up.

8  Dynamic Scenario Input (Beta)
New architecture allows SOSS to be a part of a distributed simulation environment. In the future aircraft data will be provided by a component unrelated to SOSS and added to the simulation at any time. (As opposed to all aircraft being loaded from a text file and known before simulation begins). Currently this capability is simulated by Scenario Manager that creates new aircraft objects only
within a specified forecast window of that aircraft’s PTime. As soon as aircraft “lifespan” is exhausted within the simulation (departure has crossed a fix or arrival has reached its gate and finished disembarking) it will be removed and cleaned up from simulation. Current use of this feature include significantly increasing simulation performance for scenarios spanning a large time window and involving a large number of aircraft. To enable this mode, the user must enable Dynamic Simulation Input and specify absolute input scenario path within SOSS configuration file. See “Dynamic Scenario Input Controls (Beta)” within configuration file for details.