eNGENUITY Technologies

Visual. Software. Solutions.

TECHNICAL OVERVIEW

FLSIM

Version 8.0.1

Overview

The description of kinematical and dynamical equations of flight, specific to different types of aircraft, is an integral part in the development process of flight simulators. Although the equations of motion have been known for a number of years, customizing them for one specific aircraft can become cumbersome and prolonged. FLSIM allows users to concentrate on tailoring the behavior of the aircraft by entering flight test or design data for the different aerodynamic coefficients instead of spending their time on writing source code for the different aerodynamic equations.

Evolutionary rapid prototyping is now becoming a standard in several organizations dealing with avionics development. Using reconfigurable cockpits, avionics developers, systems integrators and training organizations are providing cost-effective solutions to the total system development life cycle. For these organizations, FLSIM is a complementary software product ideally suited to stimulate the user interfaces developed using VAPS. It offers a user-friendly interface for non-programmer users, reconfigurability and flexibility for R&D environments, and reliable real-time performance for training applications.

What is FLSIM?

The new standard in low cost, high fidelity reconfigurable flight simulation.

FLSIM is a software package designed to expedite the development of low cost, high fidelity flight simulation programs. FLSIM provides a user friendly, flexible, and reliable real-time environment for flight simulation. Not only can it be used for flight training, but also to test aircraft design, as well as to test aircraft performance under controlled, simulated, conditions.

It provides an environment for configuring any fixed-wing aircraft using definable flight parameters, and for flying the resulting flight model across an Out-The-Window (OTW) scene. FLSIM provides a testing and plotting capability to enable the certification of the simulation program within a full flight simulator. It supports direct interfacing with the VAPS Run-time Environment (RE) and VAPS CCG in order to animate VAPS generated displays. Bi-directional communication also allows the user to control FLSIM through "virtual" controls (e.g. gear, speed brakes, throttles. etc.). It can also be integrated with third party products such as MatrixX.

A development environment can also be purchased. The user can append functionality not provided by the base program, or modify the default behavior and therefore achieve a higher level of customization.

Here are a few examples of the utility of the development environment:

- replacing the "flight control computer simulation module" in order to replicate the exact behavior on board a specific aircraft;
- adding the simulation for a fire control radar;
- providing an interface to custom hardware like joystick, throttle, sound generators, control loading system.

Simulation Approach

The approach taken by FLSIM for simulating an aircraft is very simple. It can involve three (3) types of users:

1. A 'Modeler' uses FLSIM to enter the aerodynamic parameters for a particular aircraft.

2. A 'Developer' could add functionality or replace existing behavior to achieve an extremely high level of fidelity.

3. The 'End User' uses the final product to fly different missions.

As can be expected, each task is suited to different interests and skills. This approach lets each 'domain expert' work in their discipline. It allows the modeler to concentrate on aerodynamics and not have to become a programmer. It allows programmers to focus on writing the code for a particular system without becoming an authority in aerodynamics.

Finally, it permits the end user to direct their attention to flying the resulting aircraft.

Flight Simulation Domain

FLSIM can be used for simulating a wide variety of aircraft driven by either piston, turbojet, turbofan thermal, or turboprop engines. The classes of aircraft that can be simulated with FLSIM include:

1. Large transport;
2. High speed fighter;
3. Small private jet; and
4. Remotely Piloted Vehicle (including UAVs and drones).

FLSIM was designed to support internal integration as well as external interfacing with user defined application software. Its internal structure makes it available as a standalone software product or as part of a complete turnkey solution for an R&D or a training environment.

FLSIM meets two basic requirements:

1. Usability: The ability to address user's requirements and ease of use.
2. Flexibility: The ability to provide solutions for a wide range of applications.

In addition, its open architecture, published data, and control interface allows the end user to incorporate FLSIM as part of their internal software configuration.

Skills Needed to use FLSIM

FLSIM allows the domain experts to concentrate on their individual disciplines. Each category requires the competence appropriate for the task.

The Modeler

The modeler communicates with FLSIM by providing the aircraft physical and aerodynamic parameters. They should have a fairly good understanding of aircraft behavior and aerodynamic terminology such as a college undergraduate course in flight dynamics. They should have a knowledge of the interaction between the aerodynamic coefficients and the aircraft response.

The Developer

The developer requires programming experience with the 'C' language as well as compilation routines specific to either the SGI or NT/2000 platforms (i.e. the 'make' utilities provided on the Unix workstations, or .dsw files from MSDEV). Although the FLSIM internal data is well structured, the developer will be required to access the appropriate data structures in real-time.

The End User

Once the simulated aircraft has been designed, it can be turned over to the end user. This person will need the basic knowledge of how an aircraft is controlled. This person could also be required to understand specific aircraft system and subsystem operation.

Hardware Requirements

FLSIM runs on Silicon Graphics Inc. (SGI) workstations with IRIX 6.5 operating system and 24 color bitplanes, and Intel Windows NT V4.0 service pack 5 and Windows 2000 workstations.

Software Requirements

FLSIM is a complete standalone software package. However, when using the development environment, it will be required to have a 'C' compiler/linker as well as the Unix 'make' utility for IRIX workstations, or the Visual C++ V6.0 service pack 3 for Windows NT and Windows 2000 workstations.

FLSIM Features

- Support for Shared Objects / DLLs.
- New scalable Turbofan Engine Model.
- Initial conditions to define:
  - Location
  - Initial fuel mass
  - Flap/gear position
  - Magnetic variation
  - Autotrim with one wing down
  - Brakes on/off
  - Throttle position, on ground and in-flight
  - Selection of tuning frequency and power status of all NAVAIDS receivers
  - All AFCS modes
  - Launching platform speed in the X/Y axis
- Test Utilities that allow to:
  - Perform comprehensive dynamic tests
  - Save all test configurations for re-use
  - Ramp any analog or discrete pilot input and control surface movement
  - Bypass the engine model and specify thrust values directly
  - Bypass the weight and balance model and enter specific values for the CG,

inertia and mass of the aircraft.

- A Plotting Utility which allows to save the plot set-up configuration, and sample data

  at various rates (i.e. at each iteration of the base frame rate, or a fraction thereof).
- Full pilot input Record and Playback function supporting, recording, playback, and resume capabilities. Playback can be finite or cyclic. Control can be taken back at the end of a playback, in order to resume flying the aircraft. A playback can also be stopped before the end in order to resume control of the aircraft.
- The units displayed to the operator in the Graphics User Interface can be customized simply be editing an ASCII file.
- Graphical curve definition for aerodynamic coefficients.
- Ablity to interpolate or extrapolate curve data.
- Complete test utility.
- Extensive plotting ability.

- Additional loads can be added to an aircraft, and jettisoned during runtime.
- Ability to integrate user defined applications.
- Standard communication interface with the VAPS Runtime-Environment (RE V5.3) or CCG.
- Real-time controller execution options which include:
  - Synchronous real-time;
  - non-real-time;
  - linked;
  - slaved to a signal/event coming from an external process.
- Complete Record, Playback and Resume capability.
- Complete Snapshot and Restore capability.
- Ground model, including steering capability.
- Advanced undercarriage model including both oleo strut and tire models
- Atmospheric model including wind gust, temperature/pressure profile completely defined by user
- Wind turbulence models including:
  - Dryden Turbulence
  - Von Karman Turbulence
  - TSO-C117 Wind Shear model
- Ellipsoidal earth model with constant rotational speed and gravitational, centripetal and coriolis effects consideration.
- Full 6 degrees of freedom, quaternion based simulation.
- Flight Management system with time prediction, cross track error, track angle error
- Automatic landing

- VOR tracking
- Autopilot system tuning tools
- ARINC 424 navigational database import capability
- Distributed Interactive Simulation (DIS) capability
- Includes a basic Out-The-Window(OTW) scene for runtime flight with generic HUD and simulation status overlay.
- Positioning of the FLSIM OTW scene anywhere on the workstation screen via an environment variable.
- Network licensing.
- Malfunctions available in the following categories:
    - Electrical System
    - Hydraulic System
    - NAVAIDS
    - Flight Instruments
    - Control Surfaces
    - Undercarriage
    - Additional Loads
    - Propulsion
    - Autopilot
    - Fuel System

- All profiles are saved in the External Data Representation (XDR) format (which is machine independent).

Supported Reference Frames

Flsim uses six different reference frames to express the equations of motions of an aircraft. These are:

- Earth Centered Geodetic (ECG)
- World Coordinate System (WCS)
- North-East-Down System (NED)
- Body System
- Stability System
- Wind System

The reference framework for the aerodynamic model can be selected among body, stability, or wind. All necessary conversions of aero data are performed automatically by FLSIM.

Simulation Models Overview

The FLSIM program is composed of the following models:

- Electrical System
- Hydraulic System
- Earth System
- Ambient System
- Winds System
- ADC System
- Weight and Balance System
- AP Logic System
- FMS Navigation System
- FMS Guidance System
- AP FCC System
- Autothrottle System
- Engine Panel System
- Engine System
- Undercarriage System
- Flight Control System
- Aerodynamic System
- EoM System
- Navaid System
- Instruments System

Electrical System

- simulates electrical power distribution

Hydraulic System

- simulates hydraulic pressure distribution

Earth System

- ellipsoid earth model.
- modifiable rotational speed about ECC x-axis.

- modifiable equatorial and polar radii.
- considers gravitational, centripetal and coriolis effects.

Ambient System

- Provides two methods of calculating ambient conditions at the aircraft position:
    - Constant Temperature Lapse Rate model (ICAO standard model).
    - Ambient Air Condition Profile model.

Wind System

- Provides two methods for calculating mean wind at the aircraft position:
    - The Von Karmen model.
    - A Mean Wind Profile.
- Provides three methods for calculating wind turbulence at the aircraft position:
    - The Von Karmen model.
    - The Dryden Model.
    - A Wind Turbulence Profile.

ADC System

- Simulates various sensors related to the monitoring of ambient conditions.

Weight and Balance System

- Computes the mass, center of gravity position and moments of inertia of each of its subsystems: airframe, fuel load, and additional loads.

AP Logic System

- Manages all engagement logic associated with the aircraft's Automatic Flight Control System (AFCS).
- Consists of two independent systems: autopilot, and autothrottle system.
- The AutoPilot (AP) provides the following modes of control:
    - Roll:
        - Heading Hold (basic mode)
        - Heading Select
        - VOR
        - Localizer
        - FMS

  - Pitch:
    - Attitude Hold (basic mode)
    - Altitude Hold
    - Glide Slope
    - FMS
  - Yaw:
    - Turn Coordination (basic mode)
    - Align
    - Rollout
- The AutoThrottle System (ATS) provides the following modes of control:
  - Speed
  - Mach
  - Retard
  - FMS

FMS Navigation System

- Computes each waypoints' distance to turn.
- Indicates to the Guidance system and the Autothrottle system the need to change altitude and speed in order to navigate towards the next waypoint.
- Provides linear deviation in the horizontal and vertical planes of the aircraft from the trajectory.
- Provides actual cross track, and course angular errors.

FMS Guidance Sytstem

- Provides new altitude and speed commands to steer the aircraft towards the desired waypoint.

AP FCC System

- Controls the aircraft according to the selected autopilot mode.

Autothrottle System

- The purpose of this system is to maintain the aircraft's speed.

Engine Panel System

- Centralizes commands coming from either pilot control devices or messages, and makes them available to the engine system

- Calculates for each engine, the power that the generator must provide for the electrical accessories.

Engine System

- The engine system simulates the following type of engines:
    - Turbojet
    - Turboprop (Basic)
    - Turboprob (Advanced)
    - Turbofan (Thermo)
    - Piston

Undercarriage System

- Controls the deployment and braking systems.
- Provides two forces systems: basic, and advanced.

Flight Control System

- Calculates new deflection of each control surface.
- Evaluation of the control law provides either a commanded deflection or a commanded rate of change.
- Simulates surface actuators.

Aerodynamic System

- Provides the aerodynamic force and moment vectors, expressed in the aircraft's body frame.
- Sum's vectors to provide the total aerodynamic force and moment vectors acting on the aircraft.

EoM System

- Considers the motion of the aircraft to be defined by the motion of its reference center of mass, which is described by Newton's second law applied to both its translational and rotational movements.
- Uses ECC system as fixed reference frame.
- Uses quaternion parametre representation to describe the transformation from the ECC frame to the body frame.

Navaids System

- Simulates the following receivers: DME, VOR, ADF, TACAN, ILS, and Marker Beacon.

Instruments System

- Simulates lag and imprecision of some flight instruments.
- Provides a stall warning indication.
- Provides access to the deflection of the control surfaces, and provides indication as to whether they are in transit.

Usability

The software is divided into four main processes:

1.	FlsimSim	the flight models and real-time controller.
2.	FlsimDe	the user interface (database editor) which controls the operation of the flight models.
3.	FlsimScene	the out the window (OTW) display
4.	FlsimVapsInt	the special interface between the simulation models and the VAPS Runtime Environment (RE) and VAPS CCG

The ***FlsimVapsInt*** process is only loaded when an interface to the VAPS RE program is required in order to activate the objects on a prototype. It does not contain any flight simulation functions.

The ***FlsimScene*** is activated when FLSIM goes into run mode.

FLSIM provides added usability by being able to run in the following configurations:

1. ***FlsimDe*** and ***FlsimSim*** executing flight simulation on the same computer.
2. ***FlsimDe*** and ***FlsimSim*** executing flight simulation on different computers.
3. ***FlsimDe*** executing alone in the database edition mode.
4. ***FlsimSim*** executing as a stand-alone flight simulation and controlled by an external application.

FLSIM provides an advanced GUI to define the parameters of the flight model, engine model, atmospheric model and waypoint trajectories.

The FLSIM User Interface incorporates the following features:

1. Rationalized user interface to group related functions and to provide an overall well organized control scheme.
2. Extensive data definition interface.
3. Data definition is supported through numerical data entry or graphical curve definition using the mouse as a pointing device.
4. Data file menus to facilitate the loading of existing data files stored in the database.
5. Basic Out-The-Window (OTW) scene for runtime flight with generic HUD and simulation status overlay. Overlay displays can be toggled ON/OFF through user selectable soft-switches.
6. Ability to import/export data to facilitate data entry.

Flexibility

FLSIM's flexibility is supported through the reconfigurability of the flight model and the ability to integrate with user defined simulation models or functions. FLSIM's core software supports standard functions that can be used as baseline for more complex applications requiring special functions.

FLSIM also provides the capability of interfacing the simulation models with external applications (e.g. Visual Scene Generator, Sound Generator) via the Ethernet Local Area Network. FLSIM can export the real-time data to external application programs using a defined data buffer (defined in the FLSIM Real-Time Data Export ICD).

Runtime Controller

The FLSIM flight simulation models are executed at a rate specified with a command line option (default is 30 Hz).

The Runtime Controller function (RTC) is responsible for ensuring that all simulation models are executed in a predefined sequence. The RTC supports synchronous, asynchronous, slaved and linked execution of the simulation modules.

In the synchronous execution mode, the RTC ensures the sequential execution within fixed time intervals (real-time simulation).

In the asynchronous execution mode, the RTC controls the sequential execution within a "free-run" mode, i.e. without any time constraint on the execution cycle. It simply waits (specified by user) msec between the end of one iteration and the start of the next iteration.

In the slaved execution mode, the RTC execution is triggered by the Unix signal SIGALRM or Windows(NT or 2000) Event sent from another process. This could mean synchronous or
asynchronous execution dependent on the external process.

In the linked execution mode, the FLSIM simulation models can be lionked into a user application that deals with custom specifications. The user program is responsible to dispatching the FLSIM models. In this case, the user function, the RTC and the models, are all merged into one process. This can be achieved by using library functions provided by FLSIM's developer package.

The RTC also supports some analysis tools to characterize the operating parameters of FLSIM. The RTC records the average iteration time, the number of overruns (in synchronous mode) and an iteration counter (frame counter).

Pilot Inputs

In addition to using the mouse and keyboard for pilot inputs, FLSIM is capable of interfacing to real hardware devices such as stick, throttle and pedals. As an option, VPI can supply a set of real hardware stick, throttle and pedals with the complementary A/D interface board. The supplied device driver handler includes a filtering functions that stabilize the A/D signals and generate "dead band" to avoid command drift when the control is not used.

Output Data

FLSIM can be configured to send the real-time data (define in an ICD document) to external processes through standard communication (Ethernet UDP/IP protocol). This output data is generated by the simulation models and stored in the Flsim Export Buffer. FLSIM supports VPI's VAPS RE/CCG standard communication channels (through the ***FlsimVapsInt*** process). The user is capable of implementing different communication standards by defining and linking application communication modules with the FLSIM program. All significant data generated by the simulation models are visible to user defined modules.

Development Libraries

Users of the FLSIM development libraries can supplement FLSIM with functionality not provided by the default models, or modify the default behavior of the software in order to further customize it. There are two methods of adding user written modules to FLSIM. They can be either statically linked to the FlsimSim program, or included in a stand alone shared library (dll on Windows). In either case, FLSIM provides a standard API though which the interface can be done. There are two types of API structures, one for system models and one for sub-system models. System models are the top level simulation entity (i.e. Electrical System, Engine System, Instruments System, etc.). Sub-system level models are part of the more complicated system which can be replaced independently of the other sub-systems. The user also has the ability to add system level models which are not part of the default FLSIM model set.

It is possible for a user module to replace any combination of any of the FLSIM simulation models described above, as well as the sub-systems they are comprised of.

The Development Libraries also enables the users to replace VPI's standard GUI with a custom developed control interface to incorporate special features addressing specific requirements.

In addition, the FLSIM simulation models can be linked into a user application that deals with custom specifications. An example of this would have the user writing an application in Ada and providing an Ada binding mechanism to the FLSIM models.