The quarterly newsletter for JSBSim, an open source flight dynamics model in C++

Volume 1, Issue 1

15 April 2004

Back of the Envelope

Key Features of JSBSim:

- Open Source
- Multi-Platform
- Object-Oriented C++
- Data driven using XML format configuration files
- Versatile Flight Control System model
- Aerodynamics modeled using component buildup
- Can be used by itself or to drive a larger simulation application with graphics

Inside this issue:

Introduction: What is JSBSim?

JSBSim is essentially a program that simulates the operation and dynamics of an aircraft. It is a program written in the C++ programming language, it is open source, freely available and redistributable under the terms of the GPL. JSBSim compiles with any robust C++ compiler, including g++ (packaged with Linux and CygWin, etc.), Borland C++, Microsoft Visual C++ (and others), and under any operating system including the various Linux, Microsoft Windows, and Macintosh operating systems.

JSBSim of course provides a math model for integrating the rigid body equations of motion, but it also features a way to build up a flight control system and/or autopilot of any complexity by assembling strings of components such as filters, gains, and switches.

JSBSim models the aerodynamics of an aircraft using the component buildup method, where the various contributions of lift (for example) are summed from the coefficient values (which are calculated dynamically at runtime). Also modeled are several kinds of engines: rocket, piston, and turbine. Each can be connected to an appropriate thruster, such as a propeller or nozzle.

JSBSim was designed from the start to be useful to students of aerospace engineering. For this reason (among others) JSBSim was given a scripting capability a few years ago. Users can script simulation runs to be made in

The name "JSBSim" became the official name for the project when the coordinator (whose initials are JSB) couldn't come up with a name. The project coordinator for FlightGear finally said, this is what *we're* going to call it internal to FlightGear. The name stuck.

batch mode (using JSBSim in a standalone mode), with the results being plotted automatically using an auxiliary JSBSim tool, "simplot". That's the topic of a separate column, though! *-JSB*

Basis for JSBSim EOM Math Model

The math model that JSBSim uses for the equations of motion is based on several references. The basic EOM is based on a sort of flat-earth set of equations that would be familiar to students using aeronautical textbooks such as Etkin and McCormick, etc. This flat-earth approach is modified to account for a spherical earth, and the converging lines of longitude as one gets closer to the poles. This approach works quite well for low speeds (neither coriolis nor centrifugal accelerations are currently modeled).

Other key documents used include Richard McFarland's paper on the simulation approach taken at NASA Ames Research Center. See the JSBSim web site for links to other key documents used in developing JSBSim, and also of general simulation interest. *-JSB*

News Items

- JSBSim is undergoing a major change — a major improvement. It has been known for a long time that the JSBSim landing gear model exhibits oscillations while at rest on the ground. Indeed, the zero-velocity landing gear problem is well-known to many simulation engineers, and is not unique to JSBSim. Looking at this problem led to a decision to refine the timestepping / integration methods used by JSBSim to propagate the state of the simulated aircraft. New JSBSim developer Mathias Fröhlich of Tübingen, Germany is a mathematician who is very familiar with the area of timestepping, and he is developing an approach to implement a much better EOM integration capability for JSBSim. This is to be the biggest reorganization that JSBSim has ever seen.
- Project Coordinator Jon Berndt will be presenting a paper about JSBSim at the annual AIAA (American Institute of Aeronautics and Astronautics) Modeling and Simulation Conference in August. This year the conference is in Rhode Island.
- In addition to the timestepping changes being developed, work is also progressing on auxiliary tools to use with JSBSim. One of those is called simplot, which is a quick and dirty plotting tool that is built on the DISLIN plotting library (see: http://www.linmpi.mpg.de/dislin/) and most recently the SimGear simulation library (see: www.simgear.org). The EasyXML feature of SimGear is being used to parse XML files for simplot. *-JSB*

The list of aircraft currently modeled for JSBSim includes those listed below. Note that many of these aircraft models are currently alpha only—they are very much works-in-progress:

- X-15
- C-172 (many flavors)
- T-38
- Space Shuttle (landing only)
- X-24b
- T-6 Texan
- F-16
- Boeing B-737
- ...

What is Digital DATCOM?

Wouldn't it be great if there was a program that could supply the aerodynamic coefficients that help drive JSBSim for any unique aircraft—or even for any given configuration? This is what Digital Datcom does—and it is a freely available application, originally produced by McDonnell Douglas under a USAF contract. From the Digital Datcom manual: "Digital Datcom calculates static stability, high-lift and control device, and dynamic-derivative characteristics using... The computer program also offer a trim option that computes control deflections and aerodynamic data for vehicle trim at subsonic Mach numbers."

One place you can get Digital Datcom is here: http://instructional1.calstatela.edu/cwu/me454/DatCom/ Apart from that you can make a request to Wright Patterson AFB. You can also purchase a CD with many useful aeronautical programs on it from Public Domain Aeronautical Software at www.pdas.com.

We plan to make more use of Digital Datcom in the future for JSBSim. *-JSB*

"This is to be the biggest reorganization that JSBSim has ever seen."

The JSBSim Web Site: www.jsbsim.org

The JSBSim project has a web site that acts as a resource for developers and users. The latest source code can be downloaded there as a tar archive, or checked out directly from CVS—our configuration management tool. Also featured on the web site are links to many online papers that are useful to simulation modelers and engineers.

One very important (and underutilized) feature of the web site is the ability to accept and store both feature requests and bug reports.

A new feature that is being discussed is the potential for a JSBSim aircraft model *hangar*. This would simply be a place to store complete aircraft models for use in FlightGear. This would be a package of files including the JSBSim-format flight model file and the instrument panels and aircraft 3D model. The idea is that a user would be able to download the file and simply unzip it in-place in a *load-n-go* operation. Stay tuned. *-JSB*

In Depth: The Visual Reference Point

It is important to point out the differences in the coordinate frames and the point of origin used for defining the particular flight model for an aircraft, and the way the aircraft is visually modeled in a 3D modeling program for display. The Visual Reference Point (VRP) is a concept which has only recently been developed. As such, it may be refined, but we present here a discussion of the VRP and the various coordinate systems used by JSBSim.

The flight dynamics model (FDM) determines where in the world the aircraft is – and by "aircraft", it is really the aircraft center of gravity (CG) that is being tracked. The FDM cares about modeling where the aircraft CG is, and the 3D model wants to be in the correct spot in the world, too, but how does one decide where to place the visual model? Can one simply take the FDM-reported CG and place the aircraft origin (0,0,0) point there? No, because the various aircraft modelers might use for a coordinate frame origin any point they see fit when creating their aircraft model. There is no universal convention for that.

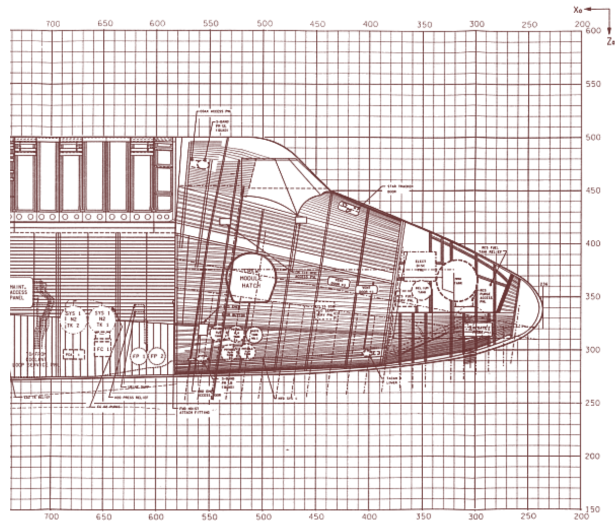
FlightGear developers discussed a possible convention. The idea is that the forward most center portion of the aircraft would be the MODEL REFERENCE POINT ("MRP" hereafter - not to be confused with the JSBSim AERODYNAMIC REFERENCE POINT). If the FDM can report the real world position of the MRP (and it can), the scene code on the FlightGear side can place the aircraft very nicely where it is supposed to be.

This is a slight problem for the FDM, though. The FDM has to make sure that it knows where the MRP *is in relation to the CG*. It is very important to remember that the FDM reports the lat/lon/alt of the aircraft CG. As fuel burns off the aircraft CG moves. So, the vector to the MRP will change with time. It's not a big issue, just a little work for the FDM.

Some Definitions

Aerodynamic Center: The idea of the aerodynamic center is similar to the idea of the center of gravity. It is the location on the aircraft through which the total lift and drag can be said to act, just as the center of gravity is the point through which the total weight acts.

Each part on the aircraft has its own aerodynamic center. In the subsonic regime, the aero center of the wing airfoil section is generally near the 0.25 chord point. But it moves aft as the aircraft increases speed into the transonic regime, typically as far back as the 0.5 chord.



This diagram illustrates the structural frame for the space shuttle orbiter. The X coordinates ("station") increase aft. The Z coordinates (waterline) increase upwards. The Y coordinates ("buttline") complete the right-hand system and increase right.

Aerodynamic Reference Point (ARP): Think about all the sources of aerodynamic pitching moment. The largest of those are the wing and the horizontal tail. That due to the tail comes largely from the tail lift multiplied by the tail moment arm. But how do we define the tail moment arm? The aerodynamic center of the wing seems like a natural choice, but doesn't really work since it moves in flight. So typically a point on the wing is arbitrarily chosen to be the moment arm zero or reference point. That's the point that has been dubbed the ARP. By convention, that point is typically along the 0.25 chord line on the wing. Spanwise, it is typically defined to be at the spanwise location of the MAC or mean aerodynamic chord. The MAC is often computed using:

$$c_{bar}/c_{root} = 2/3 * (1 + \lambda + \lambda^2)/(1 + \lambda)$$

where λ is the wing taper ratio, c_{tip}/c_{root}. Once this length is computed, the spanwise location can be found by finding the point on the wing which has that chord.

In the design phase, this point needs to be chosen early and all CFD and tunnel data reduced using it.

Structural Frame: This is the manufacturer's frame of reference used to define locations of items on the aircraft. These items would include the center of gravity, the locations of all the wheels, the pilot eyepoint, point masses, thrusters, etc. The items in the JSBSim configuration file are located using this frame. In this frame the X-axis increases from the nose towards the tail, the Y-axis increases from the fuselage out towards the right (when looking forward from the cockpit), and of course the Z-axis then is positive upwards.

(Continued on page 4)

"The idea of the aerodynamic center is similar to the idea of the center of gravity. It is the location on the aircraft through which the total lift and drag can be said to act, just as the center of gravity is the point through which the total weight acts."

The quarterly newsletter for JSBSim, an open source flight dynamics model in C++

Highlighted References

The first aircraft modeled by JSBSim was the hypersonic North American Aviation X-15 research aircraft. Here are some of the best reference materials covering that program:

Online:

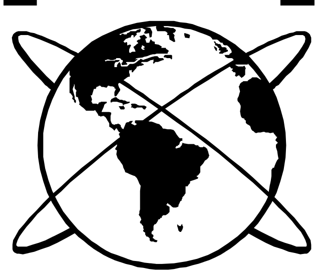
www.hq.nasa.gov/office/pao/History/x15/pubs.html

Movies: www.dfrc.nasa.gov/Gallery/Movie/X-15/index.html

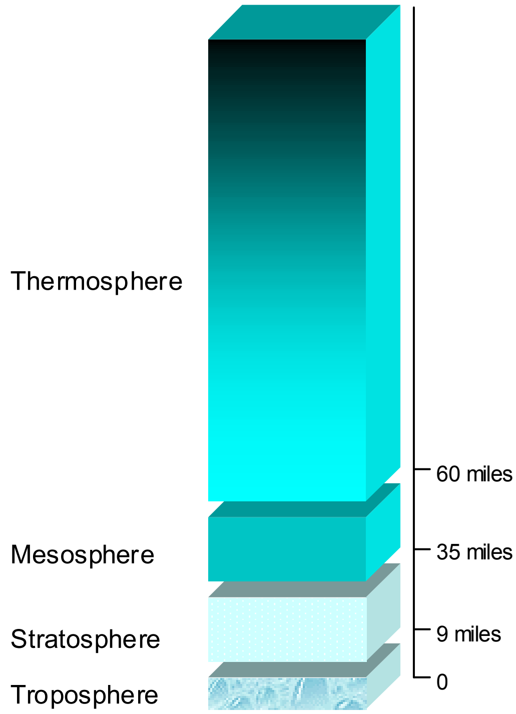
Published Books:

"Hypersonic: The Story of the North American X-15" by Dennis R. Jenkins, Tony Landis, Scott Crossfield (Foreword), William H. Dana (Foreword) ISBN: 158007068X March 2003

Visit us on the web at: www.jsbsim.org



Modeling the Atmosphere



JSBSim can use either an externally-defined atmosphere model or its own model, based on the 1976 Standard Atmosphere. Support is being added for two additional atmosphere options, the MSIS-00 model and a Martian model. [Note: MSIS is an acronym for Mass Spectrometer Incoherent Scatter.] For most tropospheric uses the 1976 Standard Atmosphere will be the best option, and it is the default. JSBSim's version of this model also includes optional wind, turbulence and temperature deviation.

Two of the drawbacks of the Standard model are a fixed tropopause and a loss of accuracy above the stratosphere. A better model for the upper atmosphere is the MSIS series of models, the newest being MSIS-00. This model was developed by Mike Picone, Alan Hedin and Doug Drob, and uses satellite drag and accelerometer data to model the atmosphere "all the way up". The model accounts for the dissociation of gases in the upper atmosphere, and can include the effects of magnetic flux. This should be useful for trans-atmospheric simulations. As far as tropospheric simulation goes, this model provides another benefit: the atmospheric conditions are also based on latitude. This means you can simulate an intercontinental flight, New York to Hong Kong for example, and not have to provide your own temperature deviations along the way. See http://nssdc.gsfc.nasa.gov/space/model/atmos/about_atmos.html for more information.

The Martian atmosphere provides a simple model of the pressure, density and temperature of the Martian atmosphere. *-DC*

(Continued from page 3)

Typically, the origin for this frame is near the front of the aircraft (at the tip of the nose, at the firewall, or in front of the nose some distance); the X-axis is typically coincident with the fuselage centerline and passes through the propeller hub (thrust axis).

Note that the origin can really be anywhere for a JSBSim-modeled aircraft, because JSBSim internally only uses the distances between the CG and the various objects – not discrete locations themselves.

Body frame: As used in JSBSim the body frame is similar to the structural frame, but rotated 180 degrees about the Y axis, with the origin coincident with the CG. This is the frame where the aircraft forces and moments are summed and the resulting accelerations are integrated to get velocities.

Stability frame (or wind axes): This frame is similar to the body frame, except that the X axis points into the relative wind *-JSB*

Next Issue:

"Back of the Envelope" is a new communication tool written for a wider audience than core JSBSim developers, including instructors, students, and other users. The articles featured will likely tend to address questions and comments raised in the mailing lists and via email. If you would like to suggest (or even author) an article for a future issue, please email the editor at: jsb@hal-pc.org.

Some possible topics for future issues includes:

- The Property System
- JSBSim Configuration Files in XML
- Integrating the Equations of Motion in JSBSim
- Scripting JSBSim runs
- Post-processing Data Output