A Journal for the Creation and Refinement of a JSBSim Aircraft Flight Model

Jon S. Berndt

In order to help those who wish to make a JSBSim flight model, but who don't have formal aeronautical instruction, this journal will serve as a guide for how I approached the problem for a chosen aircraft, the Beech 99 airliner. I chose this aircraft because there is a good set of important aircraft and flight characteristics available in several documents. [Note: an overview of the JSBSim flight model creation process was presented in the JSBSim newsletter, "Back of the Envelope", Volume 1, Issue 2.]

Figure 1 Beech 99 Airliner (Photograph © Bob Garrard , used with permission)

My first task was to collect the aircraft information. I did a search using the NASA Technical Report Server search engine at: http://ntrs.nasa.gov. The search phrase was "Beech 99". I found several documents and one was available as a PDF file. The one I used is:

> Stability and control derivative estimates obtained from flight data for the **Beech 99** aircraft
> *Tanner, R. R.; Montgomery, T. D.*
> *NASA Center for AeroSpace Information (CASI)*
> *NASA-TM-72863; H-1081 , 19790401; Apr 1, 1979*
> Lateral-directional and longitudinal stability and control derivatives were determined from flight data by using a maximum likelihood estimator for the **Beech 99** airplane. Data were obtained with the aircraft in the cruise configuration and with one-third flap deflection. The estimated derivatives show good agreement with the predictions of the manufacturer.
> Accession ID: 79N20134
> Document ID: 19790011963

Next, I searched for an FAA Type Certificate Data Sheet (TCDS) at this site:

http://www.airweb.faa.gov/Regulatory_and_Guidance_Library/rgMakeModel.nsf/

A careful search led to "TYPE CERTIFICATE DATA SHEET NO. A14CE" for the aircraft.

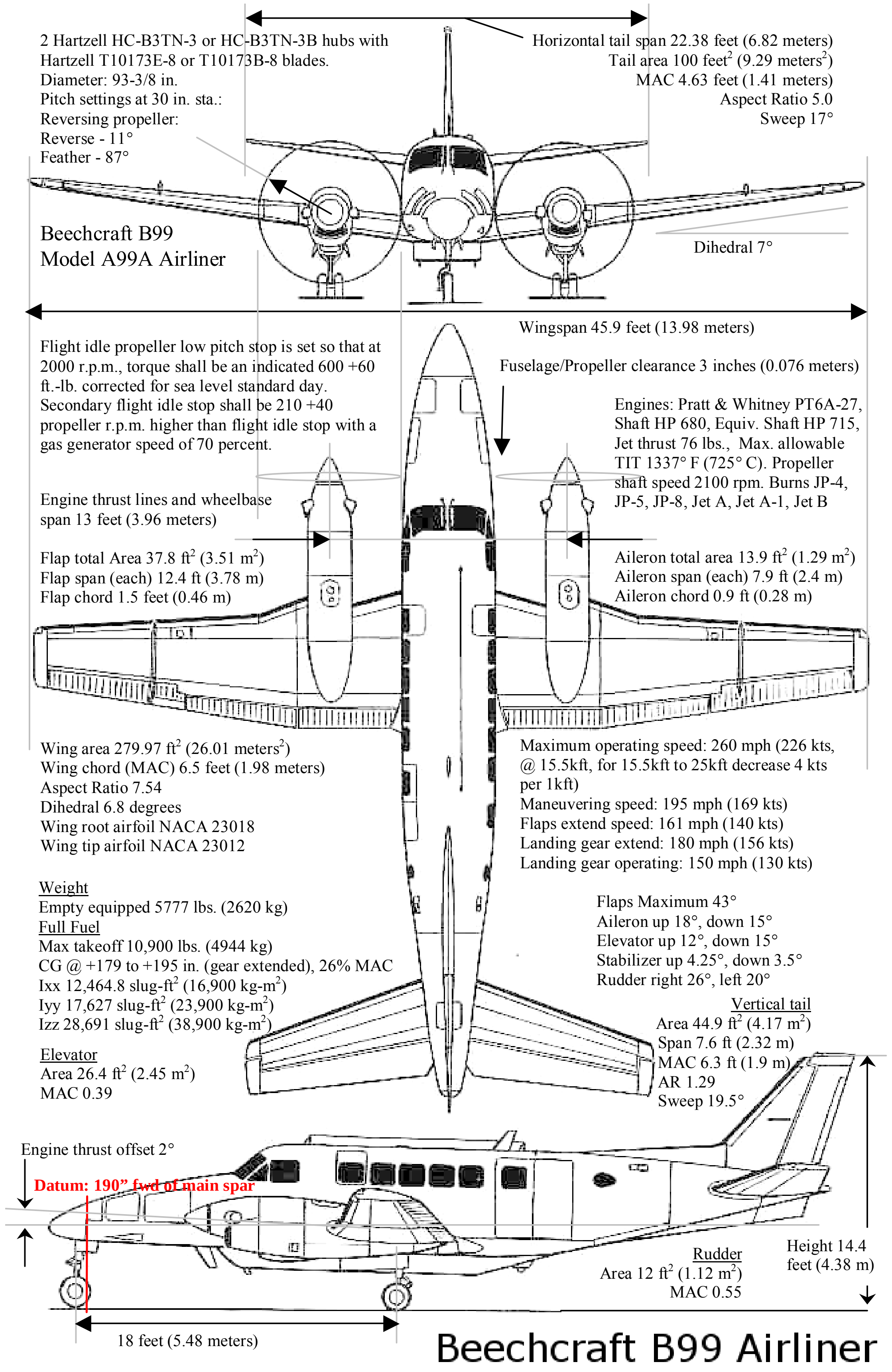
These documents will help refine a model created by the Aeromatic tool. If you are unable to find stability and aerodynamic information about the aircraft you are interested in, try finding a comparable aircraft. For instance, the CV-880, B747, and C-5 all have data published in *Aircraft Handling Qualities Data*, by Heffley and Jewell[i]. The report contains data that shows how closely (or not) various aerodynamic parameters match among similar aircraft types. For instance:

Aircraft	$C_{L\alpha}$	$C_{m\delta e}$	$C_{D\alpha}$	C_{lr}	$C_{l\delta a}$	$C_{n\beta}$
CV-880	4.28	-0.586	0.14	0.146	0.046	0.128
B747	4.5	-1.2	0.23	0.15	0.012	0.15
C5-A	5.0	-0.9	0.20	0.13	0.026	0.10

Note: The CV-880 and B747 are low-wing aircraft; the C-5 is a high-wing aircraft. Flight condition: 20,000 feet, mach 0.6.

It's also important to have a good 3-view drawing for placement of landing gear, pilot viewpoint, etc. There are a lot of good sites around. Use a search engine to find images. This site is one I've found to be particularly good:

http://membres.lycos.fr/wings2/3vues/3vues.html



2 Hartzell HC-B3TN-3 or HC-B3TN-3B hubs with Hartzell T10173E-8 or T10173B-8 blades.
Diameter: 93-3/8 in.
Pitch settings at 30 in. sta.:
Reversing propeller:
Reverse - 11°
Feather - 87°

Horizontal tail span 22.38 feet (6.82 meters)
Tail area 100 feet^2 (9.29 meters^2)
MAC 4.63 feet (1.41 meters)
Aspect Ratio 5.0
Sweep 17°

Wingspan 45.9 feet (13.98 meters)

Flight idle propeller low pitch stop is set so that at 2000 r.p.m., torque shall be an indicated 600 +60 ft.-lb. corrected for sea level standard day. Secondary flight idle stop shall be 210 +40 propeller r.p.m. higher than flight idle stop with a gas generator speed of 70 percent.

Fuselage/Propeller clearance 3 inches (0.076 meters)

Engines: Pratt & Whitney PT6A-27, Shaft HP 680, Equiv. Shaft HP 715, Jet thrust 76 lbs., Max. allowable TIT 1337° F (725° C). Propeller shaft speed 2100 rpm. Burns JP-4, JP-5, JP-8, Jet A, Jet A-1, Jet B

Engine thrust lines and wheelbase span 13 feet (3.96 meters)

Flap total Area 37.8 ft^2 (3.51 m^2)
Flap span (each) 12.4 ft (3.78 m)
Flap chord 1.5 feet (0.46 m)

Aileron total area 13.9 ft^2 (1.29 m^2)
Aileron span (each) 7.9 ft (2.4 m)
Aileron chord 0.9 ft (0.28 m)

Wing area 279.97 ft^2 (26.01 meters^2)
Wing chord (MAC) 6.5 feet (1.98 meters)
Aspect Ratio 7.54
Dihedral 6.8 degrees
Wing root airfoil NACA 23018
Wing tip airfoil NACA 23012

Maximum operating speed: 260 mph (226 kts, @ 15.5kft, for 15.5kft to 25kft decrease 4 kts per 1kft)
Maneuvering speed: 195 mph (169 kts)
Flaps extend speed: 161 mph (140 kts)
Landing gear extend: 180 mph (156 kts)
Landing gear operating: 150 mph (130 kts)

Weight
Empty equipped 5777 lbs. (2620 kg)
Full Fuel
Max takeoff 10,900 lbs. (4944 kg)
CG @ +179 to +195 in. (gear extended), 26% MAC
Ixx 12,464.8 slug-ft^2 (16,900 kg-m^2)
Iyy 17,627 slug-ft^2 (23,900 kg-m^2)
Izz 28,691 slug-ft^2 (38,900 kg-m^2)

Flaps Maximum 43°
Aileron up 18°, down 15°
Elevator up 12°, down 15°
Stabilizer up 4.25°, down 3.5°
Rudder right 26°, left 20°

Vertical tail
Area 44.9 ft^2 (4.17 m^2)
Span 7.6 ft (2.32 m)
MAC 6.3 ft (1.9 m)
AR 1.29
Sweep 19.5°

Elevator
Area 26.4 ft^2 (2.45 m^2)
MAC 0.39

Engine thrust offset 2°

Rudder
Area 12 ft^2 (1.12 m^2)
MAC 0.55

Figure 2 Beech 99 A99A Airliner Specifications (compiled by Jon S. Berndt 2006)

It should be noted that the collection of information at the beginning - the "farming" of data from the available sources – is very important. The better the information you have to start with, the better will be your resulting flight model.

The next step after locating the specific information is to use Aeromatic and create an initial model based on a few inputs. The Aeromatic tool can be found at the JSBSim web site via a link, or more directly at www.jsbsim.org/aeromatic2.html. The Aeromatic tool is well documented and its use is outside the scope of this article. However, if you have problems using that tool, please provide some information about your difficulties either in the JSBSim developer mailing list, or via the bug-reporting tool at the JSBSim web site.

Refining the Configuration File

Using the data one has found, each section of the aircraft configuration file can be refined. The File Header section can be modified to include the author's name and contact information, references can be listed, etc.

Metrics

The metrics section contains measurement information that can be taken from technical documents or drawings, but it's usually extracted from a combination of the two. The drawing included in this document was refined to include positioning information. The coordinate frame used to locate everything is in the so-called "structural frame". Every JSBSim aircraft configuration file has positioning information defined in the structural frame: the landing gear, the CG, the engine location[s], etc. The structural frame is oriented with the X-axis positive towards the tail (aft), the Y-axis positive towards the right (from a pilot's perspective seated inside the aircraft) and the Z-axis completing the right-handed system, positive upwards. The actual origin of the structural frame is not so important, because JSBSim is really only concerned about the *relative* placement of the CG, landing gear, engines, etc. So, when a structural frame for a specific aircraft is known, that frame can be used. When the frame is not known, the user can choose where the origin is. The centerline of the aircraft at the nose or the firewall is a commonly chosen location for the coordinate frame origin. A tool such as "gimp" (www.gimp.org) can be used to measure the relative distances between various parts of an aircraft. Using the FAA-issued TCDS and the drawing in Figure 2, fairly accurate estimates of CG location, landing gear location, etc. can be made in a structural frame. Also for this aircraft, the datum (origin of the structural coordinate frame) is given in the TCDS as being 190 inches ahead of the main spar centerline.

Figure 3 Beech 99 takes off in snow, flaps down (photograph ©Thomas Klein, used with permission. See www.flightphoto.net)

For this aircraft, the following locations were determined:

Aerodynamic reference point: +190 inches
Pilot eyepoint: +130.0, -18.0, +36.0 inches
Visual Reference Point: -25.0, 0.0, 0.0 inches (tip of nose of aircraft)

The wing dimensions can be adjusted in the configuration file given the actual information found. See Figure 2.

<u>Mass and Balance</u>

The mass and balance information follows the metrics section. This section contains information about moments of inertia, center of gravity location, etc. The information entered in the configuration file should be for the <u>empty weight</u> configuration. Quite often, the moments of inertia and center of gravity location are given for a fueled and loaded aircraft. Determining the mass and balance numbers in this case can be an iterative process. Once the fuel tanks are defined and located, and the content is established in the configuration file, actually running JSBSim with the initial configuration file can be used to find out what the total loaded values are. If the simulation is run for a second or two, and there is an <output> section in the configuration file, the mass_props logging can be turned on and the moments of inertia checked.

Using the information from the TCDS, and the landing gear placement in the drawing, the location of the CG can be estimated. The aft-most placement of the CG must be located such that the aircraft will not tip over backward onto its tail, of course.

The initial guess for the empty weight CG location is +190.0 inches.

Other elements specified in the metrics section include the vertical and horizontal tail moment arm. These items are specified for use in calculating response to turbulence. They are not required at this time, but may be determined from the drawing and added.

<u>Ground Reactions</u>

The ground_reactions section follows the mass_balance section in the configuration file. It contains information about wheels (or skids) and structural contact points for the aircraft. For wheels or skids the information provided in the configuration file includes:

- Location (actually, the contact point of the wheel or skid with the ground, unloaded)
- Spring and damping coefficients for the strut
- Rolling, sliding, and static friction coefficients
- Steering, retraction, and braking characteristics

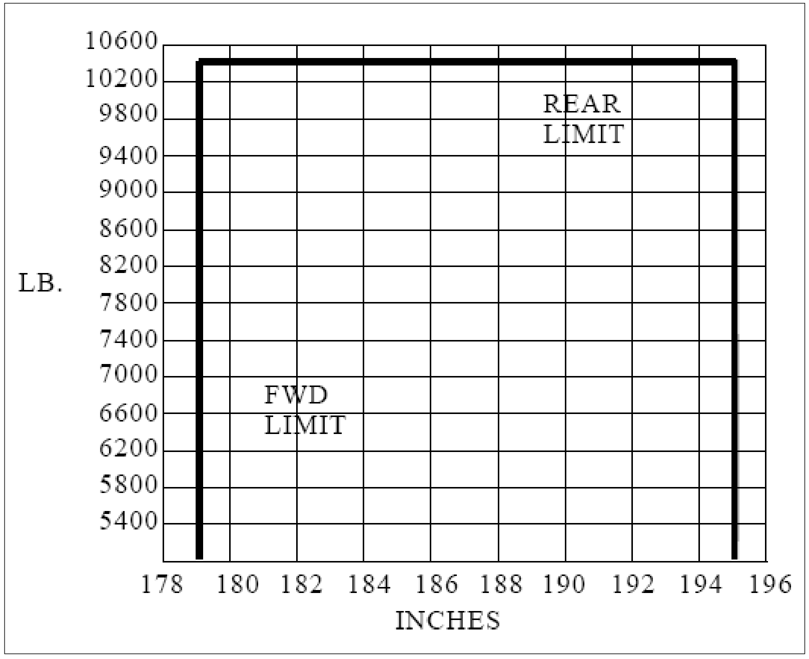


Figure 4 Center of Gravity envelope for the Beech A99A

The location can be found from Figure 2. Given the datum chosen, and the fact that the CG is stated (in the TCDS) to have a range back to x = +195 inches, we know that the landing gear must be more aft than that. Figure 2 seems to show that the main landing gear is at less than 205 inches aft of datum. We can estimate that the landing gear is at 205 inches and call that "good enough". For lateral placement of the main landing gear, we can be exact, because the landing gear are mounted in the turbine nacelle and we know the placement of the nacelle centerline in Y. The left nacelle to right nacelle span is (from Figure 2) 13 feet, so the placement of the left gear in Y is -78 inches, and the right gear is at Y = +78 inches. For the vertical placement, we know that the propeller 2.3 meters in diameter, and that the ground clearance of the propeller tip is 0.34 meters. That's 58.7 inches. We also guesstimate that at full load the aircraft compresses the main gear 6 inches, so that when the main gear are unloaded, they extend in Z to -64.7 inches.

As for the frictional characteristics of tires, the rolling friction depends on the type of tire. According to the book, *The Mechanics of Pneumatic Tires*[ii], the rolling friction for radial ply tires is in the neighborhood of 0.013-0.015 for all velocities, increasing slightly or not at all with velocity. Bias ply tires have a rolling friction coefficient in the neighborhood of 0.016-0.017 at low speed, increasing to 0.020-0.022 at a speed of 100 mph.

The tire friction model in JSBSim is at this time fairly simple. The static friction present in the configuration file shows a static and dynamic friction parameter. The static friction coefficient represents the ability of the tire to grip the surface when not sliding – that is, when the relative motion between the tire contact surface and the runway is very near zero. The dynamic friction coefficient represents the sliding friction coefficient. Typical values for static and dynamic friction coefficients are 0.8 and 0.5, respectively. More information about tire friction can be found in references iii and iv.

In the vertical direction, characteristics of the strut need to be entered. A strut consists of a spring and a damper. The spring and damper are both at this time assumed to be linear, first order. That is, the force exerted by the spring is directly proportional to the distance that it is deflected. The damper force is a function of the velocity of the compression of the strut (the sink rate of the aircraft landing gear). These quantities are not something typically found in any available literature. However, quite plausible estimates can be made. When the aircraft is at rest on the runway, each landing gear strut will deflect a certain amount (d_Z), and that amount can be calculated by knowing the weight (W_n) that the strut "n" supports and the spring constant (K_S) for that strut: $d_Z = W_n/K_S$. For a large aircraft, we might assert that each strut is compressed 12 inches from its unloaded state as the aircraft sits on the runway. If the strut is determined to support 80,000 lbs at rest, then the spring constant should be 80,000 lbs/ft. The damping coefficient is often estimated to be somewhere between 10%-30% of the spring constant value. In this case, we might first try a value of 20,000 lbs/ft/sec.

Finally, the strut definition states whether or not it is retractable, what the maximum steering angle is, and which braking group (if any) the strut belongs to.

For reference, here is the definition for a strut for the Beech 99:

```
  <contact type="BOGEY" name="LEFT_MAIN">
   <location unit="IN">
     <x> 205.0 </x>
     <y> -78.0 </y>
     <z> -64.7 </z>
   </location>
   <static_friction>  0.80 </static_friction>
   <dynamic_friction> 0.50 </dynamic_friction>
   <rolling_friction> 0.02 </rolling_friction>
   <spring_coeff unit="LBS/FT"> 8000 </spring_coeff>
   <damping_coeff unit="LBS/FT/SEC"> 2000 </damping_coeff>
   <max_steer unit="DEG">20</max_steer>
   <brake_group>LEFT</brake_group>
   <retractable>1</retractable>
 </contact>
```

Propulsion

The propulsion section states which engine[s] and thruster[s] will be used to power an aircraft. Location and orientation information is given, as well as which fuel tanks an engine may draw from. The fuel tanks are also located and defined in this section. Specific engine information is given in the engine specification file – likewise for the thruster (propeller or nozzle).

The engine and thruster are both chosen by entering the name of the engine file (minus the ".xml" extension) as the value of the name attribute. For example, the PT6A-27 engine – defined in the PT6A-27.xml file, is selected as follows:

```
   <engine file="PT6A-27">
```

The path searched for the file is first the engine/ subdirectory in the JSBSim root directory, and secondly, the engine/ subdirectory under the specific aircraft directory.

The coordinate frame that the placement of the engine and thruster take place in is the structural frame, as before. In the case of the Beech 99, the engines are placed symmetrically on each wing, 78 inches out from the fuselage centerline (see Figure 2). The engine is placed at X=120 inches, Y=78 inches, and Z=0 inches.

The orientation of the engine is parallel to the aircraft, except that the engine is pitched up 2 degrees – a positive pitch angle.

The thruster that an engine uses is defined inside the engine definition – it is "owned" by the engine definition. Like the engine, the thruster is referenced by the use of the filename (minus the .xml extension).

The specification for the first engine/propeller arrangement for the Beech 99 is shown:

```
  <engine file="PT6A-27">
    <location unit="IN">
      <x> 120 </x>
      <y> -78 </y>
      <z>   0 </z>
    </location>
    <orient unit="DEG">
      <pitch> 2.00 </pitch>
      <roll>   0.00 </roll>
      <yaw>   0.00 </yaw>
    </orient>
    <feed>0</feed>
    <thruster file="direct">
      <location unit="IN">
        <x> 120 </x>
        <y> -78 </y>
        <z>   0 </z>
      </location>
      <orient unit="DEG">
        <pitch> 2 </pitch>
        <roll>  0 </roll>
        <yaw>   0 </yaw>
      </orient>
    </thruster>
  </engine>
```

The engine is seen to draw from the first fuel tank (numbered starting with 0). The fuel tank is also defined in the propulsion section:

```
  <tank type="FUEL" number="0">
    <location unit="IN">
      <x>  205 </x>
      <y> -135 </y>
      <z>    0 </z>
    </location>
    <capacity unit="LBS"> 300.00 </capacity>
    <contents unit="LBS"> 150.00 </contents>
  </tank>
```

Autopilot and Flight Control

The autopilot and flight control sections of the configuration file describe perhaps one of the most interesting parts of JSBSim capabilities. The autopilot section is a duplicate of the format of the flight control section that follows it, and is optional, but if present (either inline or by reference) it is executed before the flight control section. Having the autopilot section separate also allows for easier reuse of autopilots between aircraft.

The flight control section will be discussed, but also applies for the autopilot section. The basic format of the flight control section of the configuration file for a JSBSim aircraft flight model is:

```
<flight_control name="aircraft_name">

  <!-- interface property declaration[s] -->
  <property> {property_name} </property>
  …

  <!-- sensor definition[s] -->
  <sensor>
    …
  </sensor>
  …

  <!–chennel definition[s] --->
  <channel name="channel_name">

    … component definitions …

  </channel>
  …

</flight_control>
```

Alternatively, with either the autopilot or the flight control section, a file may be referenced, instead, that contains the relevant definition:

```
<flight_control name="aircraft_name" file=”filename”>
```

The filename attribute is specified using only the root filename without the “.xml” extension, and the file would be searched for in the same directory as the aircraft is found in.

Within the flight control section are interface property declarations, sensor definitions, and one or more "channel" sections. Interface property declarations are optional, and in fact are usually not called for. Interface properties were conceived to accomodate the case where an aircraft is designed for use with JSBSim and FlightGear together, but for some reason (perhaps for testing purposes) it is desired to use standalone JSBSim to make a scripted run for the aircraft. Some properties specified as inputs to some flight control components may actually be defined within FlightGear, e.g. switch states defined in an instrument panel definition file. If the components that are defined on the FlightGear side are specified as interface properties, the rest of the flight control specification need not be adapted when running JSBSim standalone.

Often in the flight control laws defined for an aircraft, or in the autopilot definition, the value of a parameter sensed by the aircraft is used to direct the flight controls in some way. The list of sensed parameters typically used in flight control systems include body rates (pitch, roll, and yaw) and position, aerodynamic parameters such as angle of attack (alpha) and sideslip angle (beta), etc. These items are communicated to the flight control system through the use of various sensors. JSBSim can model "perfect" sensors (that is, actual environment values, unaffected by noise, etc.) by simply using the value in question directly – right from the equations of motion. In the real world, sensors are not perfect. Inaccuracies can creep into the values that are sensed through bias, noise, drift, failure, quantization, etc. To make simulated flight control models more realistic (or to simulate and output a quantized data stream) one can create sensors that process parameters and "rough them up" a bit, resulting in values closer to "real world" values. Here is the format of the sensor component:

```
<sensor name="name">
  <input> property </input>
  <lag> number </lag>
  <noise variation="PERCENT|ABSOLUTE"> number </noise>
  <quantization name="name">
    <bits> number </bits>
    <min> number </min>
    <max> number </max>
  </quantization>
  <drift_rate> number </drift_rate>
  <bias> number </bias>
</sensor>
```

Sensors aren't used in simple aircraft definitions, so they will not be covered here.

Channels serve to further group together components of a particular purpose. For instance, pitch channel components would be grouped together, as would yaw, and roll components. The flap control components are also often grouped together. There is not necessarily a right and wrong way to group components together – it is for user convenience, and for a future GUI editor. If desired, ALL components in the flight control system could be grouped into a single channel.

The set of components now modeled includes the following: Switch, Deadband, Summer, Gain, Aerosurface scale, Scheduled gain, Integrator, Lag filter, Lead-lag filter, Washout filter, Second order filter, and Kinematic. The idea behind the JSBSim components is that there will be a string, or chain, of components (again, possibly grouped together via the channel tag), each one producing an output that is available for a subsequent (or simply, *other*) component to reference or use as input. The first component in a string normally takes as input a control input, such as a joystick input.

There are some features that are common to all components: input, clipto, and output. The gain components have a gain associated with them. The summer components have more than one input. While the flight control specification for some aircraft can be quite complex, the default flight control laws defined for the Beech 99 by Aeromatic are quite simple. The pitch channel is defined as follows:

```
<channel name="Pitch">

 <summer name="Pitch Trim Sum">
    <input>fcs/elevator-cmd-norm</input>
    <input>fcs/pitch-trim-cmd-norm</input>
    <clipto>
      <min> -1 </min>
      <max>  1 </max>
    </clipto>
 </summer>

 <aerosurface_scale name="Elevator Control">
    <input>fcs/pitch-trim-sum</input>
    <range>
      <min> -0.35 </min>
      <max>  0.30 </max>
    </range>
    <output>fcs/elevator-pos-rad</output>
 </aerosurface_scale>

</channel>
```

There are only two components that comprise the pitch channel control laws for this aircraft. The first, a summer, takes elevator command and pitch trim (both values range from -1 to +1) and sum them, clipping the result to +/-1. The second component takes the output from the first component and maps the input into the range -0.35 to +0.30. The “Elevator Control” component also causes the elevator position property (fcs/elevator-pos-rad) to be set with the output value of the component.

A word here can be said about property names. How does one know what the property name is for an input? A list is provided in the Appendix for properties that are always available. For some parameters, properties are created on-the-fly, such as for flight control components. As an example, the first component in the pitch channel is named, “Pitch Trim Sum”. When the component is created, a property is created with a name based on the string provided in the name attribute of the component with the following process: the letters are all made lower-case, the spaces are replaced with a “/” character, and the string is prefixed with “fcs/”., resulting in the property name, “fcs/pitch-trim-sum”. In the second pitch channel component, one can see that the first component is taken as input.

Aerodynamics

The aerodynamic forces and moments are modeled using the classic coefficient buildup method. The lift, drag, and side forces, and the pitch, roll, and yaw moments need to be calculated to determine the flight path that an aircraft will take. Many factors contribute to each of the forces and moments acting on an aircraft. For instance, lift is a function of the inherent lift of the aircraft at zero degrees angle of attack, the lift due to a change in angle of attack from zero degrees, the lift due to a deflection of the horizontal stabilizer (pitch command), the lift due to pitch rate, the lift due to flap deflection, etc. Likewise, for the other five axes, the total force or moment is the sum of the individual contributions.

The aerodynamic definition for a JSBSim aircraft flight model is comprised of six sections corresponding to the forces and moments about three axes each (for a total of six degrees of freedom). In each section, any number of functions are present that define individual contributions to a force or moment. Additionally, “global” functions may be defined at the beginning of the aerodynamics section, outside of any axis section, for later use by another function.

A function in JSBSim may be composed of common arithmetic and/or trigonometric operations on one or more operands. MathML, the XML markup language for mathematical constructs, inspires the format of the JSBSim function but the JSBSim function is much simpler. A cue is also taken from AeroML[1], the proposed standard under consideration by AIAA for exchange of simulation models.

As an example, let's examine the lift force due to angle of attack (alpha). We know that increasing the angle of attack increases lift – up to a point. Lift force is the product of dynamic pressure (qbar), wing area (S_w), and lift coefficient (C_L). In this case, the lift coefficient is determined via a lookup table, using alpha as a lookup index into the table:

```
<function name="aero/coefficient/CLalpha">
  <description>Lift_due_to_alpha</description>
  <product>
    <property>aero/qbar-psf</property>
    <property>metrics/Sw-sqft</property>
    <table name="CL">
      <independentVar lookup="row">aero/alpha-rad</independentVar>
      <tableData>
        -0.20 -0.720
         0.00  0.240
         0.22  1.300
         0.60  0.664
      </tableData>
    </table>
  </product>
</function>
```

What the above function results in is for "CLalpha" to be calculated each pass through the aerodynamics system calculations in JSBSim. The value of the function is the product of qbar, wing area, and lift coefficient as determined by the lookup table. For instance, if alpha (in radians) is 0 degrees, the lift coefficient is 0.240.

So, where do we get this information about the Beech 99? Aeromatic has created for us a boilerplate configuration file populated with plausible values for many aerodynamic terms. However, there are a lot of refinements that can – and *should* - be made. First of all, the document mentioned at the beginning of this article contains specific information about the aerodynamic qualities of the aircraft. Let's look at how to *tweak* a JSBSim aerodynamics specification for a specific aircraft.

As mentioned, JSBSim uses the coefficient buildup method for calculating aerodynamic forces and moments on an aircraft. We have a nice report for the Beech 99 showing both the manufacturer's estimates and flight test data for the aerodynamic quantities:

Longitudinal coefficients (i.e. xz-plane symmetric coefficients, lift, pitch, drag)

(1) $$C_N = C_{N_\alpha}\alpha + C_{N_{\delta e}}\delta_e + C_{N_0}$$

(1a) $$C_L = C_{L_\alpha}\alpha + C_{L_{\delta e}}\delta_e + C_{L_0}$$

(2) $$C_m = C_{m_\alpha}\alpha + C_{m_q}\frac{q\bar{c}}{2V} + C_{m\delta_e}\delta_e + C_{m_0}$$

(3) $$C_D = C_{D_0} + KC_L{}^2$$

[1] At this time, AeroML is known as DaveML. See http://daveml.nasa.gov for more information.

Lateral coefficients (i.e. side force, roll, yaw)

(4) $C_Y = C_{Y_\beta}\beta + C_{Y\delta_a}\delta_a + C_{Y\delta_r}\delta_r + C_{Y_0}$

(5) $C_l = C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r + C_{l_0}$

(6) $C_n = C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r + C_{n_0}$

The above equations are for (in order), normal force coefficient, pitch moment coefficient, drag coefficient, side force coefficient, rolling moment coefficient, and yaw moment coefficient. There is no lift coefficient specified. For low alpha, the normal and lift coefficients are very close.

The manufacturer's estimates for the aerodynamic force and moment components are as follows:

(7)	$C_{L_\alpha} = 0.096 + 0.151 \cdot T_c$	(per degree; no flaps)
(8)	$C_{L_\alpha} = 0.101 + 0.151 \cdot T_c$	(per degree; 1/3 flaps)
(9)	$C_D = 0.0275 + 0.0625 \cdot C_L{}^2$	
(10)	$C_{m_{\alpha,1/4}} = 0.010 + 0.010 \cdot T_c$	(per degree)
(11)	$C_{m\delta_e} = -0.034 - 0.032 \cdot T_c$	(per degree)
(12)	$C_{m_{\dot\alpha}} = -43.1$	(per *radian*)
(13)	$C_{m_q} = -3.40$	(per *radian*)
(14)	$C_{n_\beta} = 0.014 - 0.0015 \cdot T_c$	(per degree)
(15)	$C_{l_\beta} = -0.0023$	(per degree)
(16)	$C_{Y_\beta} = -0.010$	(per degree)
(17)	$C_{n\delta_r} = -0.0014$	(per degree)
(18)	$C_{l\delta_r} = 0.00017 - 0.000022 \cdot \alpha$	(per degree)
(19)	$C_{Y\delta_r} = 0.0026$	(per *radian*)
(20)	$C_{n_r} = -0.20 - 0.197 \cdot \alpha$	(per *radian*)
(21)	$C_{l_r} = 0.05 + 0.623 \cdot \alpha$	(per *radian*)
(22)	$C_{Y_r} = 0.394$	(per *radian*)
(23)	$C_{n_p} = 0.0079 - 0.762 \cdot \alpha$	(per *radian*)
(24)	$C_{l_p} = -0.515$	(per *radian*)
(25)	$C_{Y_p} = -0.199 - 0.385 \cdot \alpha$	(per *radian*)
(26)	$C_{l\delta_a} = 0.0027$	(per degree)
(27)	$C_{n\delta_a} = -0.0015$	(per degree)

The subscripts *p*, *q*, *r*, refer to rates about the *x*, *y*, *z*, axes; δ_x (where *x* is *a*, *r*, *e*) refers to the deflection of the aileron, rudder, or elevator, respectively; *l*, *m*, *n*, refer to moments about the *x*, *y*, *z*, axes; *α*, *β*, refer to angle of attack and angle of sideslip; *L*, *D*, and *Y*, refer to lift, drag, and side force. T_c refers to thrust coefficient, and is defined as follows:

$$(28) \qquad T_c = \frac{Thrust}{\overline{q}S}$$

These coefficients and derivatives are not all necessarily the last word on aerodynamic modeling for the Beech 99 – the data comes from testing that was done for a particular purpose that is not necessarily supposed to result in a full aerodynamic database for simulation modeling. So, we are left with data obtained within a fairly narrow flight envelope. We can look at the coefficient and derivative data and make adjustments on a one-by-one basis. But, not only are there adjustments that can be made to the set of coefficients, but additional ones that can be added in.

Let's consider $C_{L\alpha}$ first. The symbol $C_{L\alpha}$ is shorthand for $\delta C_L/\delta_\alpha$, or *the change in lift coefficient with a change in alpha*. This is just the slope of the lift coefficient curve (see Fig. 5 for an example). The lift curve slope for an ideal 2D airfoil section is 2π (per radian). For a cambered airfoil, the lift curve does *not* pass through zero – that is, there is an inherent lift to the airfoil even at zero angle of attack. Furthermore, a real wing (i.e. *not* a 2D airfoil section) is not as efficient as an airfoil, and the slope of the lift curve will be somewhat less than 2π. Additionally, a real wing will eventually stall at a particular angle of attack, perhaps around 12-16 degrees. The information given in the Beech 99 report is valuable, however, some adjustments will have to be made to account for stalling. The value of $C_{L\alpha}$ defines a slope – a line that goes on forever!

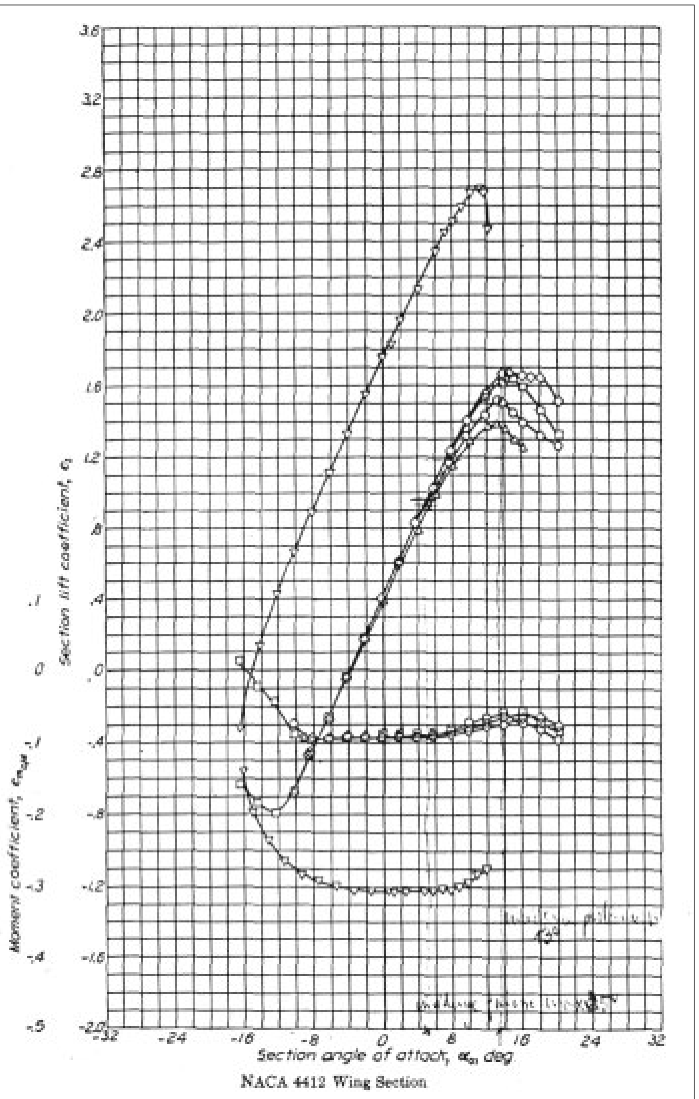


Figure 5 Lift curve for NACA 4412 Wing Section (2D)

Knowing that the lift attributable to alpha is complex – it's not a straight line – we might instead consider creating a lookup table or function that includes the behavior at stall. Unfortunately, little or no information is given to us about stall in the document, but some searching suggests 73 kts (123 ft/sec) as the stall speed.

Also, what about the post-stall behavior? Modeling post-stall and very high alpha regimes is effectively beyond the scope of this discussion, but suffice it to say we would like to provide *plausible* behavior in all regimes where possible. In this case, we can look to a study done by Sandia Labs some years ago[v]. The study resulted in data being collected that showed the lift coefficient as it varied with angle of attack for symmetric airfoils of varying thickness, across the entire alpha range of 0 to 180 degrees. Figure 6 (next page) shows the lift coefficient for an airfoil for a full range of alpha. The thickness of the airfoil represented by the data in Fig. 6 is 12%. The blended-airfoil wing used on the Beech 99 is 18% thick at the root and 12% thick at the tip. We will assert that we can use the data for a 15% thick airfoil as representative of the average characteristic of the wing.

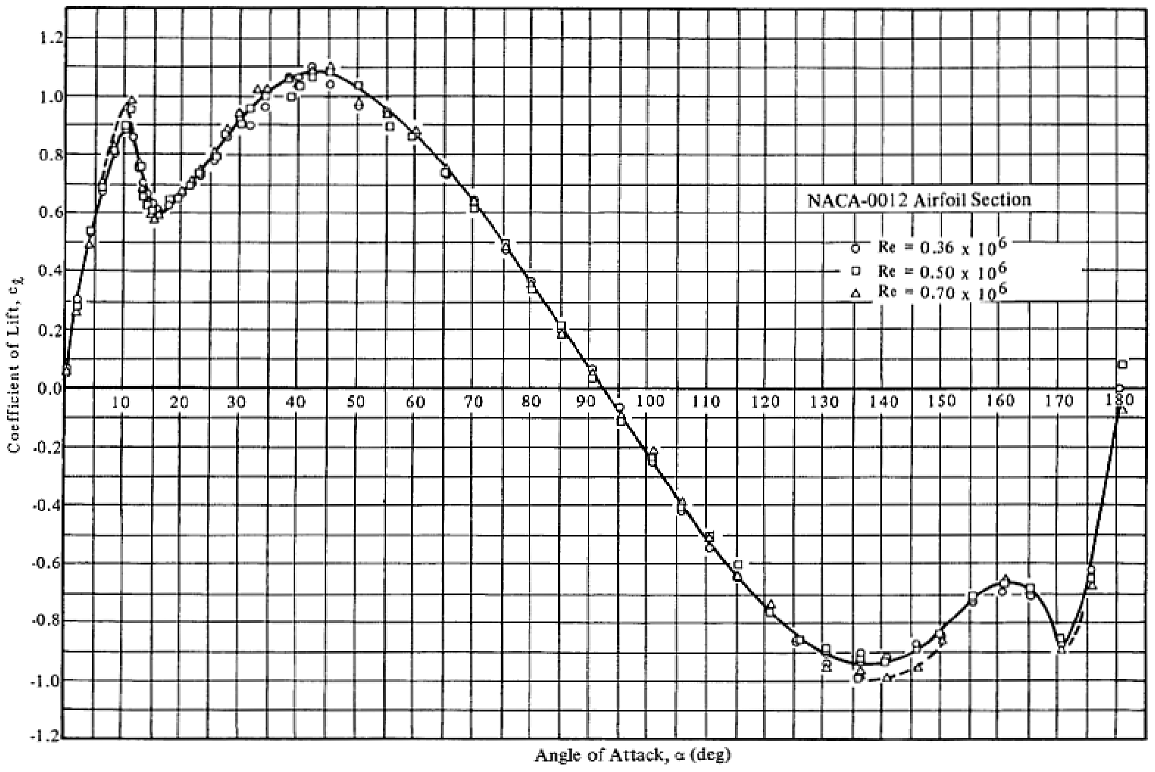


Figure 6 The lift coefficient for a 12% thick symmetric NACA airfoil from 0 to 180 degrees angle of attack.

Not including thrust effects at the moment, the lift coefficient of the wing can be written as:

Lift coefficient = 0.939 * {table value} + (0.25 * {flap position degrees} * 0.017) + 0.3

The lift force is just the above coefficient multiplied by qbar and area:

Lift force = qbar * wingarea * Lift_coefficient

The equation was developed by asserting that the effect of camber on the airfoil data effectively adds 0.3 to the lift coefficient, that for a 3-D wing the curve must be scaled (0.939), and that adding a flap setting in degrees (multiplied by 0.017 to convert it to a radian measurement) multiplied by 0.25 gives the increase in lift coefficient from flaps.

The JSBSim-ML representation of the lift of a wing using the 15% thick Sandia data and information from “Theory of Wing Sections”, NACA wing data, etc. is shown as:

```
<function name="aero/force/CLalpha">
  <description> Lift force due to alpha </description>
  <product>
    <property>aero/qbar-psf</property>
```

```
<property>metrics/Sw-sqft</property>
<sum name=”aero/coefficient/CLalpha”>
  <value> 0.3 </value> <!-- Lift curve slope camber shift -->
  <product>
    <value> 0.25 </value> <!-- flap setting curve shift -->
    <property> fcs/flap-pos-deg </property>
    <value> 0.017 </value> <!-- convert degrees to radians -->
  </product>
  <product>
    <value> 0.939583333 </value> <!-- Lift curve scaling value -->
    <table>
      <independentVar lookup="row">aero/alpha-deg</independentVar>
      <tableData>
      -180   0.0000
      -170   0.8500
      -160   0.6350
      -150   0.7700
      -140   0.9800
      -130   0.8500
      -120   0.6700
      -110   0.4500
      -100   0.1850
       -90  -0.0900
       -80  -0.3650
       -70  -0.6300
       -60  -0.8750
       -50  -1.0200
       -40  -1.0350
       -30  -0.8550
       -20  -1.3325
       -19  -1.3608
       -18  -1.3897
       -17  -1.4136
       -16  -1.4233
       -15  -1.4136
       -14  -1.3825
       -13  -1.3300
       -12  -1.2591
       -11  -1.1749
       -10  -1.1000
        -9  -0.9900
        -8  -0.8800
        -7  -0.7700
        -6  -0.6600
        -5  -0.5500
        -4  -0.4400
        -3  -0.3300
        -2  -0.2200
        -1  -0.1100
         0   0.0000
         1   0.1100
         2   0.2200
         3   0.3300
         4   0.4400
         5   0.5500
         6   0.6600
         7   0.7700
         8   0.8800
         9   0.9900
```

```
           10   1.1000
           11   1.1749
           12   1.2591
           13   1.3300
           14   1.3825
           15   1.4136
           16   1.4233
           17   1.4136
           18   1.3897
           19   1.3608
           20   1.3325
           30   0.8550
           40   1.0350
           50   1.0200
           60   0.8750
           70   0.6300
           80   0.3650
           90   0.0900
          100  -0.1850
          110  -0.4500
          120  -0.6700
          130  -0.8500
          140  -0.9800
          150  -0.7700
          160  -0.6350
          170  -0.8500
          180   0.0000
        </tableData>
      </table>
    </product>
  </sum>
 </product>
</function>
```

This specification should produce a fairly smoothly changing lift coefficient in a wide range of conditions. Note that this really is a force calculation – the lift coefficient has been multiplied with qbar and the area of the wing.

This function defining the lift force on the wing due to angle of attack is but one contribution to the total lift force on the aircraft. When the elevator is moved, it also changes the lift force of the tail, which results in a pitch moment on the aircraft – the main purpose of the elevator – but the change in lift force must be taken into account in the total lift force calculation.

---

[i] Aircraft Handling Qualities Data, Heffley, Jewell, 1972, NASA CR-2144. This document can be found at www.jsbsim.org/fslinks.html.

[ii] *The Mechanics of Pneumatic Tires"*, http://media.wiley.com/product_data/excerpt/19/04713546/0471354619.pdf

[iii] *A Study of the Mechanical Properties of Modern Radial Aircraft Tires*, NASA/TM-2003-212415, Robert Daugherty

[iv] *Friction Characteristics of 20x4.4, Type VII, Aircraft Tires Constructed With Different Tread Rubber Compounds*, NASA TN D-8252, Dreher and Yager

[v] *Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack For Use In Aerodynamic Analysis of Vertical Axis Wind Turbines*, R.E. Sheldahl, P.C. Klimas, SAND80-2114 (http://infoserve.library.sandia.gov/sand_doc/1980/802114.pdf)