YSFlight Aircraft Performance Analysis: Calculating Angle of Attack Using Euler Rotations

By: Decaff_42

Abstract—**Many aircraft performance parameters are reliant on the aircraft's Angle of Attack, such as Lift and Drag coefficients along with YSFlight stall parameters. In the interest of analyzing the performance of YSFlight aircraft an Excel tool was developed to extract meaningful aircraft flight data from a .yfs file. In order to calculate Angle of Attack no matter what the airplane is doing, Euler Rotations are used.**

I. Introduction

The extraction of Angle of Attack (AoA) and Angle of Side Slip (AoS) data is needed when analyzing the lift and drag characteristics of an aircraft. In addition, this data can be used to analyze rudder and elevator effectiveness. To calculate these angles, the aircraft's velocity vector must be calculated relative to the aircraft's Body Reference Frame (ARF). This poses a problem because .yfs data only includes the aircraft's position and orientation relative to the Global Reference Frame (GRF) used by YSFlight. Therefore, derivative parameters such as the global Velocity vector components must be extracted from the changing position data.

Another issue that needs to be overcome is that the GRF of YSFlight is a Left-Hand Cartesian Coordinate System. This causes problems translating the GRF velocity vector of the aircraft into the ARF. Additionally the aircraft orientation angles are not recorded in the traditional Right-Hand Rule. The information needed will be translated into a standard Right-Hand Cartesian Coordinate system and while this is done a standard aviation ARF and the Tangent Plane Reference Coordinate System GRF will be adopted for calculations.

II. Aircraft Reference Frame

The standard ARF for airplanes has the positive x-axis pointed towards the nose along the aircraft's center-line. The positive y-axis points towards the right wing, while the positive z-axis points directly downward. You can see the axis and angle notation in Figure 1.

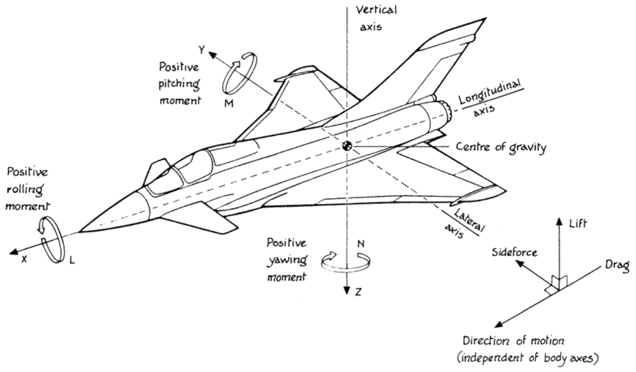

Fig. 1. A Diagram of Aircraft Coordinates and Angles.[1]

III. Tangent Plane Reference Coordinate System

The Tangent Plane Reference Coordinate System is used in a wide variety of applications such as GPS. Even YSFlight uses a version of it, although only in a loose sense. The core idea is that wherever you are on Earth, there is a tangential plane that intersects the earth at one point. This point is now your origin. The x-axis points due north, the y-axis points due east, and the z-axis points directly down towards the center of the earth. The similarities between this and the standard aviation ARF are quite apparent and coincidentally the YSFlight coordinates provided in the .yfs file simply need to be re-arranged and the sign of the altitude flipped in order to translate to the Tangential Plane GRF.

As you can see in Figure 2, the tangential plane could be moved anywhere on the surface of the earth, but to an aircraft flying close to the origin, it would still look the same. In YSFlight there is no curvature of the earth, so in essence the entire YSFlight Map is on this tangential plane. This makes it easy to work with.

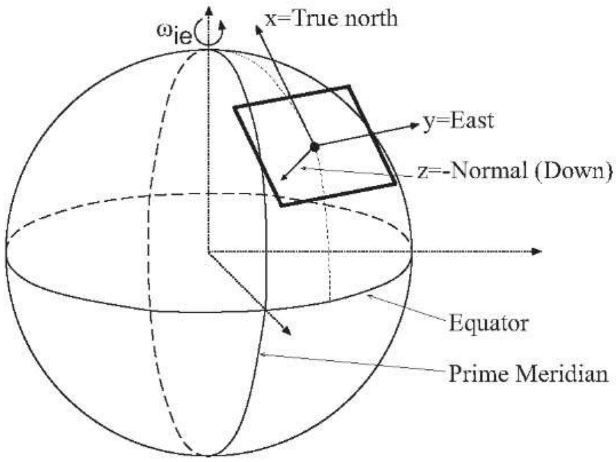

Fig. 2. Tangent Plane Reference Coordinate System On Sphere.[2]

IV. .yfs File Data

The .yfs file records a lot of information about each flight. Data stored in a .yfs file comes in four lines per data entry as shown below:

```
0.72
200.0 332.218 74.179 -0.0 -0.0019 0.0 0.8
0 218 0 0 0 0 0 0 240 15 24 0 0 0 0 0 0 0
0
```

1) The first line indicates the time in seconds since the start of the simulation.
2) The second line covers where the aircraft is and what direction it is going. It indicates the x, y and z positions followed by the x, y, and z rotations and finished with the 'g' load on the aircraft.
3) The third line is where lots of information is stored regarding aircraft components. Some of the information is mixed into a single channel. A more detailed analysis regarding the effectiveness of the control surfaces would also pull data from this line.
4) The fourth and last line is where turret information is recorded, however if there are no turrets, then the line defaults to a single zero as shown above.

A. Transforming YSFlight GRF to Tangential Plane

For calculating AoA and AoS, we only need 7 pieces of information. Time, X Y & Z positions, and the aircraft rotation relative to the axes. The first step in using this data is to convert the X Y & Z positions from the YSFlight Coordinate System into the Tangential Plane coordinate system. The table below shows how this is done.

TABLE I
YSFlight Coordinate Conversion to Tangential Plane

New Coordinate	Old Coordinate
X	Z
Y	X
Z	-Y

B. YSFlight Angle Notation

The aircraft rotation data that is included in the .yfs file has its own notation which we will be working on. This data includes the yaw, pitch and roll of the aircraft relative to Due-North straight and level flight. All the data is provided in radians.

1) Yaw: The Yaw rotation is defined with Due-North being 0 radians and Due-South being either π or $-\pi$ radians depending on the direction you turned. If turning North to West to South, the yaw angle is positive. If turning North to East to South, the yaw angle is negative.

To adopt the standard shown in Figure 1, the following logic can be used to transform the angle to have a range of 0 to 2π radians.

```
If R_yaw > 0
    R_yaw = 2 * Pi - R_yaw
Else
    R_yaw = -1 * R_yaw
```

2) Pitch: The Pitch angle is defined as being positive above the horizon and negative below the horizon. The following logic can be used to calculate the transformed Pitch rotation.

```
If R_pitch < 0
    R_pitch = 2 * Pi + R_pitch
```

3) Roll: The Roll angle is defined as being positive with the right wing up and negative with the right wing down. The following logic can be used to calculate the transformed Roll rotation.

```
If R_roll < 0
    R_roll = 2 * Pi + R_roll
```

C. Calculating Velocity

The instantaneous velocity of the aircraft cannot be calculated. Instead the average velocity over the last time step can be calculated knowing the change in position and the time it took to do so. This will require looking at two time-steps in order to calculate the velocity at the current time. These time-steps will be indicated by the subscript (i) or $(i-1)$ to indicate the current and immediately preceding time-steps respectively. The X, Y and Z components of the velocity need to be calculated, however the formula is the same for each. Therefore the components are represented by (α) in the equation below for $i > 1$ where $i = 1$ at the $t = 0$ time-step.

$$V_{\alpha i} = \frac{\alpha_i - \alpha_{i-1}}{t_i - t_{i-1}} \tag{1}$$

The mild downside to this method is that the initial $t = 0$ velocity will be infinite as there is no previous time to subtract and having a zero in the denominator is never a good thing. The solution is to set the initial condition $(i = 1)$ to zero for all velocity vector components. While the aircraft certainly has velocity in the simulation, there is no data to get this information from the .yfs time-steps.

Another option might be to simply set the $i = 1$ velocity components to be equal to the $i = 2$ components as it is extremely unlikely for a significant velocity change to occur less than a quarter second after simulation start.

V. Euler Rotations

A common method of translating information such as vectors or positions between two coordinate systems, the second having been rotated, is by an Euler Rotation. These rotations are characterized by a rotation matrix (R_α) that involves a rotation of α radians about a particular axis as seen in equations 4 through 6. For more information on Euler Rotations, refer to Professor Strongol's Lecture on Aircraft Equations of Motion.[3]

Equation 2 shows the general equation used for the Euler Rotation where V_{1-3} represent the velocity components in the ARF and $Vx - z$ represent the velocity components in the Tangential Plane.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = R_\alpha \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{2}$$

It should be noted that R_α can be a combination of multiple rotation matrices and can be expanded slightly to represent a three-angle Euler Rotation as shown in equation 3.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = R_\phi R_\theta R_\psi \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{3}$$

The different Euler Rotation matrices are listed below. It is important to note that you cannot change the order these matrices are multiplied in, or you will change the final vector. See equations 7 through 9 to see the order these are performed in.

Rotation Matrix about x-axis:

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{4}$$

Rotation Matrix about y-axis:

$$R_\theta = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{5}$$

Rotation Matrix about z-axis:

$$R_\psi = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

While it is possible to combine the three rotation matrices into a single 3x3 matrix, doing so may result in tremendous error if the rotation matrix is not calculated properly. Experiments done in Excel VBA saw upwards of 40 percent error in velocity magnitude using combined rotation matrices taken from published sources. It was therefore necessary to perform the rotation in three stages as shown below.

$$\begin{bmatrix} V_{x1} \\ V_{y1} \\ V_{z1} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} V_{x2} \\ V_{y2} \\ V_{z2} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} V_{x1} \\ V_{y1} \\ V_{z1} \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} V_{x2} \\ V_{y2} \\ V_{z2} \end{bmatrix} \tag{9}$$

Percent error can be calculated by comparing the velocity magnitude calculated from the components in the Tangential Plane and the velocity magnitude calculated from the rotated components in the ARF. The general equation for velocity magnitude is shown below.

$$V = \sqrt{{V_x}^2 + {V_y}^2 + {V_z}^2} \tag{10}$$

VI. Combined Rotation Matrix

A combined rotation matrix takes patience to calculate, however using a single 3x3 combined rotation matrix multiplication with the trigonometric polynomials, is slightly faster to compute than using three simple rotation matrices.

A series of 12 runs with 500 seconds of YSFlight time, 8601 frames, was used to determine how much effect the combined rotation matrix made on calculation time. Using Excel VBA, a the combined rotation matrix calculations were performed 0.85 percent faster. While there are fewer Matrix calculations, there are 29 trigonometric functions to calculate for the combined rotation matrix. In comparison the 3 single-axis rotation matrices only required 12 trigonometric functions to calculate.

Below you can find the combined rotation matrix with a shorthand notation for Sines and Cosines. C_θ is the Cosine of θ while S_θ is the Sine of θ. The same applies for the other angles.

$$R = \begin{bmatrix} C_\theta C\psi & C_\theta S_\psi & -S_\theta \\ S_\theta S_\phi C_\psi - C_\phi S_\psi & S_\theta S_\phi S_\psi + C\phi C_\psi & S_\phi C_\theta \\ C_\psi C_\phi S_\theta + S_\phi S_\psi & S_\psi C_\phi S_\theta - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix} \tag{11}$$

Therefore Equation 2 can be re-written to include the combined Rotation Matrix as seen in Equation 12:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} C_\theta C\psi & C_\theta S_\psi & -S_\theta \\ S_\theta S_\phi C_\psi - C_\phi S_\psi & S_\theta S_\phi S_\psi + C\phi C_\psi & S_\phi C_\theta \\ C_\psi C_\phi S_\theta + S_\phi S_\psi & S_\psi C_\phi S_\theta - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{12}$$

VII. Calculating AoA and AoS

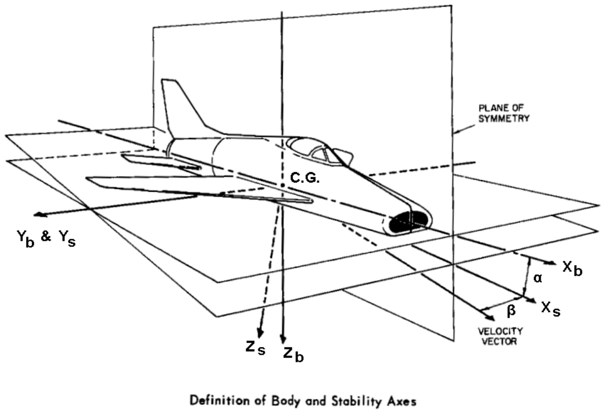

Fig. 3. AoA and AoS relative to Aircraft.[4]

Now that the velocity vector is in the ARF, AoA and AoS can be calculated using some simple trigonometry. It is important to remember that the vertical velocity component will be positive for a downward velocity and negative for an upward velocity per the ARF being used. By looking at

Figure 3, it is easy to see that simple trigonometry is all that is needed to calculated these angles from the velocity components. Typically α is used to indicate AoA and β for AoS. The equations for both are shown below.

$$\alpha = -\tan^{-1}\left(\frac{V_3}{V_1}\right) \tag{13}$$

$$\beta = \tan^{-1}\left(\frac{V_2}{V_1}\right) \tag{14}$$

VIII. Excel VBA Code

You can download a .txt version of my excel code to calculate Euler Rotations from mediafire.com. Download Here

This is a simple example of how you might write your own code. Excel VBA is used merely as an example. Translating it into another language should not be all that hard, however you should watch out for radians vs degrees when doing trigonometry as many programing languages will only use radians (Excel, Python, Matlab to name a few).

References

[1] R. Whitford *Design For Air Combat*. Jane's Publishing Company Limited, London, England, 1987.

[2] The-Crankshaft Publishing, *Reference Frame Definitions (GPS)* What-When-How. http://what-when-how.com/gps-with-high-rate-sensors/reference-frame-definitions-gps/

[3] R. Strongol, *Aircraft Flight Dynamics MAE331 Lecture 9 - Aircraft Equations of Motion-2* Princeton University https://www.princeton.edu/~stengel/MAE331Lecture9.pdf

[4] Airplane's Body, Stability and Wind Axis. http://dodlithr.blogspot.com/2011/09/airplanes-stability-axis.html