Angle of Attack and Elevator Trim Models: Setup and Operation

June 14, 2015

Until recently, MatrixPilot flight stabilization and navigation controls neglected the angle of attack of the wing of fixed wing aircraft. This lead to small errors in several of the computations, including pitch control, centrifugal compensation, and wind estimation. To improve the accuracy of these calculations, an angle of attack model was added to the computations. This document explains the setup and operation of this new feature.

Angle of attack is approximately linearly related to a parameter referred to in this document as relative wing loading, which is proportional to wing lift and inversely proportional to the square of the air speed, with scaling such that the relative wing loading at nominal cruise speed during straight and level flight is equal to 1. It is possible to estimate the angle of attack dependence on relative wing loading from flight data. The dependence that was measured during a typical flight of an HILSIM EasyStar2 is shown in Figure 1.

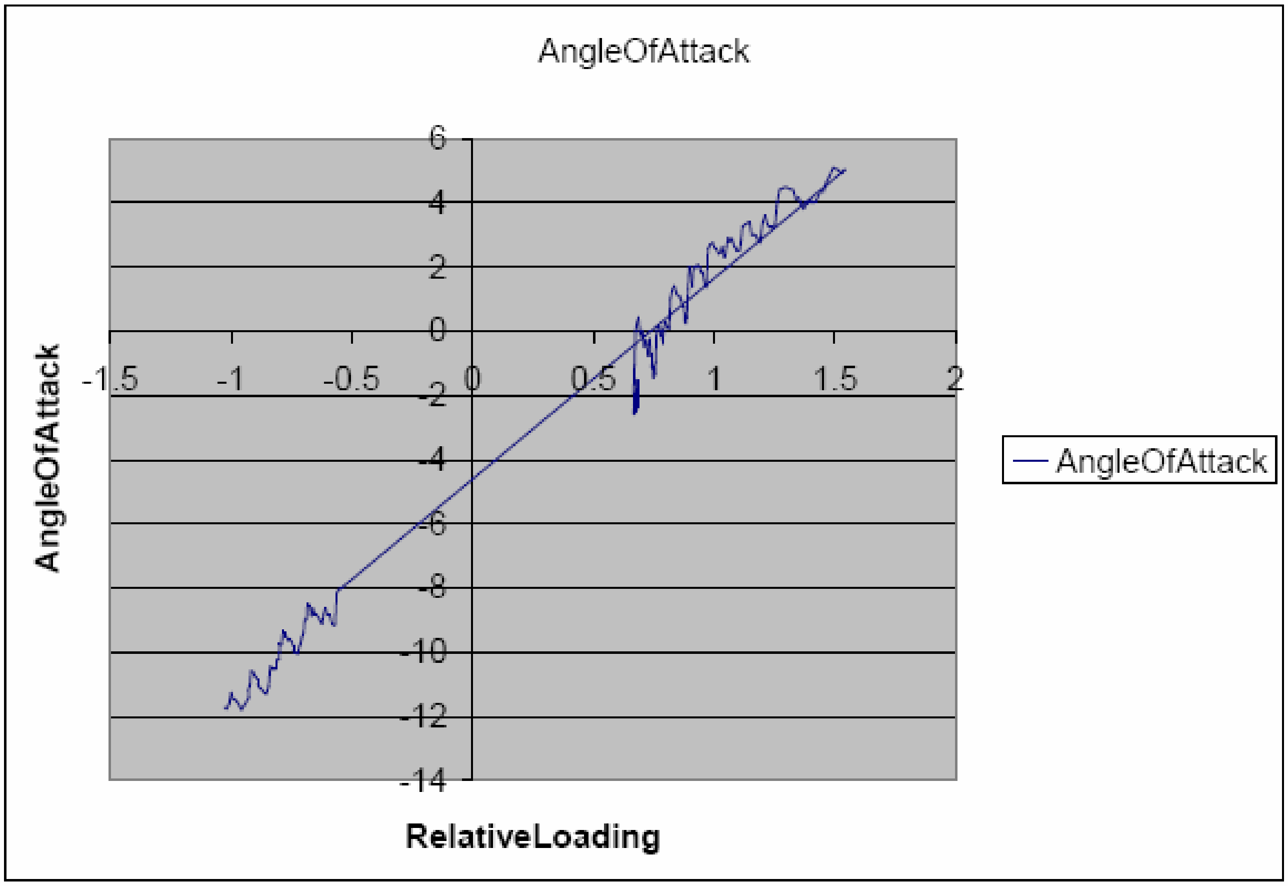

Figure 1 - Dependence of angle of attack on relative wing loading

Figure 1 was constructed by measuring pitch angle during straight and level flight at constant altitude, with both normal and inverted orientation, while airspeed was varied by adjusting elevator trim. It can be seen that the dependence is approximately linear.

It is worthwhile to note that if you plan to use the angle of attack model, you must take some care in setting the accelerometer offsets. This must be done with the UDB as level as possible with respect to the earth. The best approach is to use the roll-pitch-yaw demo to measure the accelerometer offsets on the bench while the UDB is level, and then use the option to use those specified offsets for flight rather than those recorded on power up. For example, the following are the offsets specified in one of the author's options.h files:

```
#define CUSTOM_OFFSETS
#define XACCEL_OFFSET    ( 363 )
#define YACCEL_OFFSET    ( 36 )
#define ZACCEL_OFFSET    ( -1097 )
#define XRATE_OFFSET     ( -93 )
#define YRATE_OFFSET     ( -8 )
#define ZRATE_OFFSET     ( -141 )
```

Detailed instructions for recording sensor offsets are given in the companion document, RecordingSensorOffsets.

If you would rather not use custom offsets measured with the UDB out of the aircraft, the next best thing is to measure the offsets during power up, with the UDB in the aircraft. However, you should make every effort to level the UDB with respect to the earth. In other words, regardless of which mounting orientation option you are using, during power up recording of sensor offsets, one axis of the UDB should be parallel with earth vertical, and the other two should be in the earth horizontal plane.

For best overall accuracy, the sensor offsets should be recorded for a level UDB. Not doing so is equivalent to accepting a slight misalignment between gyro and accelerometer axes, which generates small errors in nearly all of the estimation and control firmware. In other words, it is not possible to properly represent the relationships among UDB, earth, and aircraft by simply recording sensor offsets on power up. The proper approach is:

1. Record UDB accelerometer offsets with respect to vertical by measuring sensor offsets with the UDB level.
2. Mount the UDB approximately aligned with the expected airflow. The alignment does not need to be perfect with respect to pitch. That is because the UDB establishes a reference frame for pitch and angle of attack, and any offset is accounted for by the angle of attack model.
3. Use the MatrixPilot rmat values in flight to estimate angle of attack with respect to the UDB. Or, use Peter Hollands flight analysis program to extract the parameters.

Before discussing how to determine the angle of attack parameters, it is useful to first discuss a related topic: elevator trim. During HILSIM testing of the angle of attack model, it was discovered that the improvement in altitude control during inverted flight or aggressive turns was not as much as hoped for. It was realized that in addition to an angle of attack model, an elevator trim model is also needed. That is because the center of gravity of an aircraft is not normally aligned with the center of pressure. The center of gravity is usually slightly forward from the center of gravity. This is done to produce inherent pitch stability. For example, if an external disturbance pitches a plane slightly upward from a stable flight orientation, it will climb slightly and decelerate. As the airspeed drops, the torque generated by the separation of center of gravity and center of pressure drops, and the pitch decreases and compensates for the disturbance. In steady state the pitch settles to an equilibrium value.

This means that the equilibrium elevator trim depends on relative wing loading. The approximate dependence is linear. This was verified using flight data from a HILSIM EasyStar2 flight. The variation of elevator trim with relative wing loading is shown in Figure 2. Elevator trim is computed as a ratio of actual trim to maximum deflection.

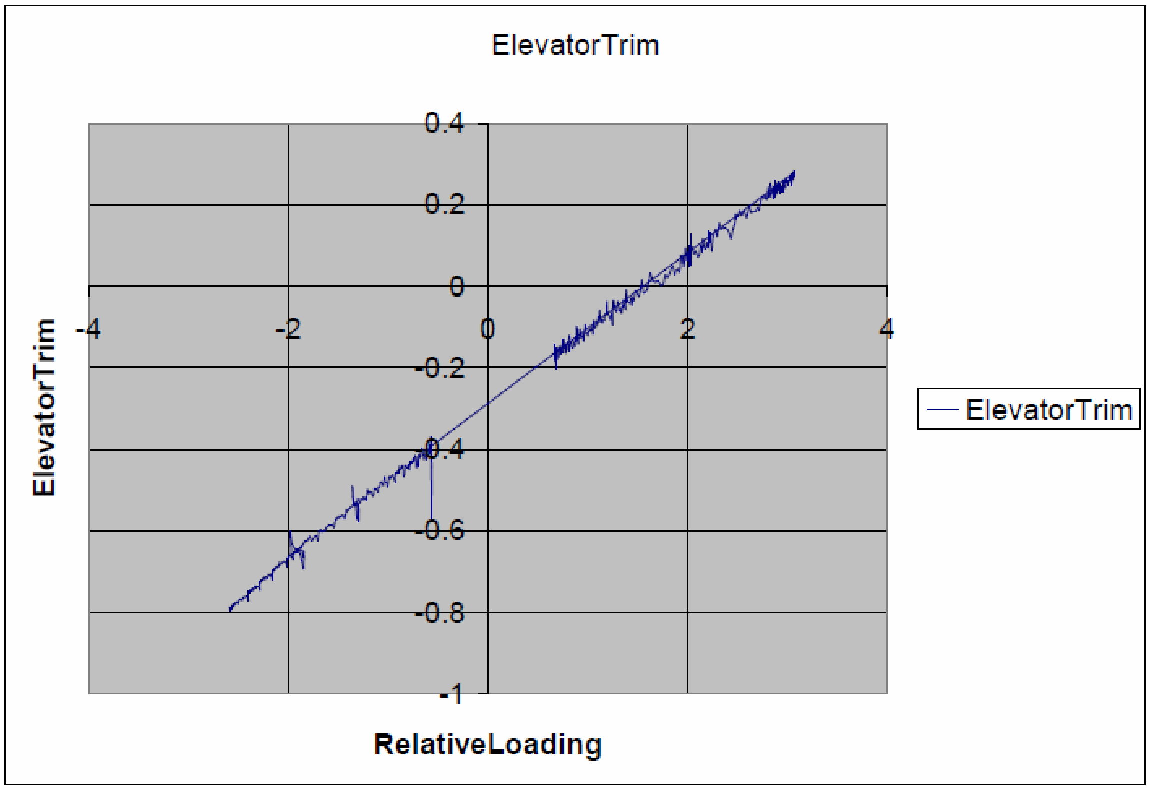

Figure 2 - Elevator trim as a function of relative wing loading

Therefore, both an angle of attack model and an elevator trim model are needed in order to improve the performance of pitch controls.

The parameters needed for the models can be obtained from flight data by varying airspeed during straight and level flight and then plotting pitch and elevator deflection as a function of relative wing loading. Some thought was given to also using data during aggressive turns, but there are a couple of issues with that approach. The angle of attack is not equal to the pitch angle during a turn. Furthermore, the elevator deflection includes the amount required to achieve the necessary pitch rate in the aircraft frame of reference. So, the best strategy is to obtain data during straight and level flight with altitude control turned on. That way, the air velocity vector is in the earth frame horizontal plane, and the pitch angle is equal to the angle of attack. Also, the elevator deflection is equal to the amount needed to trim for wing loading.

The technique to capture the data is to fly in fly-by-wire mode with altitude control turned on, and both throttle and elevator used by the controls. Allow the altitude control to stabilize after launch. Then very slowly pull back on the elevator stick to increase the angle of attack. The controls will respond by slowly climbing, reducing throttle, and reducing airspeed. Pull back very slowly. Stop when the aircraft shows signs of stalling.

If you plan to fly inverted, then you should take data in both inverted and normal orientation. If you do not plan to fly inverted, then data taken in a normal orientation is sufficient to determine the angle of attack and elevator trim models. If you do plan to fly inverted, it is a good idea to get a first approximation for the parameters in normal flight orientation, program those parameters, and then repeat another flight including inverted flight. The first approximation for the parameters will improve the pitch performance during inverted flight.

Obviously it will not be possible to fly perfectly straight during the entire process. So you will have to make some turns. Just keep them as wide and as gentle as you can.

After your flight, analyze the flight data to determine the model parameters. You can either do the analysis manually using a spread sheet, or you can use the flight analysis program, which will be discussed later. If you do the analysis manually, here are some tips for analyzing and plotting the data.

1. Angle of attack is equal to the pitch angle for normal level flight, it is equal to the negative of the pitch angle for inverted level flight. For more resolution, compute pitch from rmat[7] in the telemetry rather than using the reported values of pitch angle. Since the angle of attack is small, you can use rmat[7] directly rather than bothering to perform an inverse sine. Don't forget that rmat[7] has the opposite sign of pitch, so you will need to flip the sign of rmat[7] to get pitch. Since rmat[7] is scaled by 16384, and since 180 degrees is equal to pi radians, divide minus rmat[7] by 286 to get angle of attack in degrees for normal flight, and divide rmat[7] by 286 to get angle of attack in degrees for inverted flight.
2. For pitch trim, use the telemetry pulse width data for the PWM output to the pitch channel. To get trim values, subtract the PWM input for that channel with the sticks centered prior to takeoff. Convert from raw telemetry data to trim values by dividing by 1000. Also, you will need to take into account the polarity of the pitch channel. If you have selected servo reversing for that channel in your options.h file, you will need to flip the sign. There is no additional sign flip for normal/inverted, because the pitch trim is in the body frame of reference by definition.

3. The slope of both plots should be positive. The slope of angle of attack versus relative wing loading is always positive. The slope of pitch trim versus relative wing loading is positive as long as the center of gravity is forward of the center of pressure. If either or both of your plots comes out with a negative slope, you have probably flipped a sign somewhere in your computations.
4. If you are including angle of attack data for both normal and inverted flight, you should be able to draw an approximately straight line through all of the data. If you cannot you may have forgotten to flip the sign of one relative to the other.
5. In order for this technique to work, you should look at data only when the plane is flying approximately straight and level at constant speed, so down select data that satisfies those conditions. In particular, look at both altitude and IMUZ vertical velocity to assure that the flight trajectory is level.
6. If you do not plan to fly your plane inverted, then obviously you will be using points with positive relative wing loading. In that case, in order to draw a straight line, you will need several points, which you can get by varying airspeed by adjusting pitch trim. You will need to extrapolate the line to negative values of wing loading. Do not be concerned about the accuracy of the extrapolation, because as long as you do not fly inverted, that portion of the model will not be exercised.
7. If you plan to fly inverted, then you will have data for both normal and inverted flight. In that case, drawing the line will be easy.
8. For angle of attack and pitch trim models, you will need to enter values of angle of attack and pitch trim for relative wing loadings of plus and minus 1, which you can read from your plots.

Another approach is to use the flight analysis program to determine the model parameters by selecting that option in the flight analysis user interface. A typical plot of angle of attack as a function of relative wing loading produced by the flight analysis program is shown at the top of Figure 3, where relative wing loading is equal to the wing g loading times the square of the ratio of cruise air speed divided by actual airspeed. The data is taken from a HILSIM flight, so there is not much noise. For a real flight, there will be more noise, but the estimated parameters are reasonably accurate.

The flight analysis program will compute the angle of attack model parameters using two different methods, velocity vector and pitch angle. The method that uses velocity vector is generally more accurate, but both methods are reported. Plots are displayed and model parameters are printed in the flight analysis dialog window.

In order to produce the angle of attack plots from the flight analysis program, you will need a version of python that supports the statistical and plotting functions used to compute and plot the model parameters. If you do not have the required functions, the flight analysis program will inform you and tell you where to get them. Or, you can post your data to the MatrixPilot users group and ask for someone to process it.

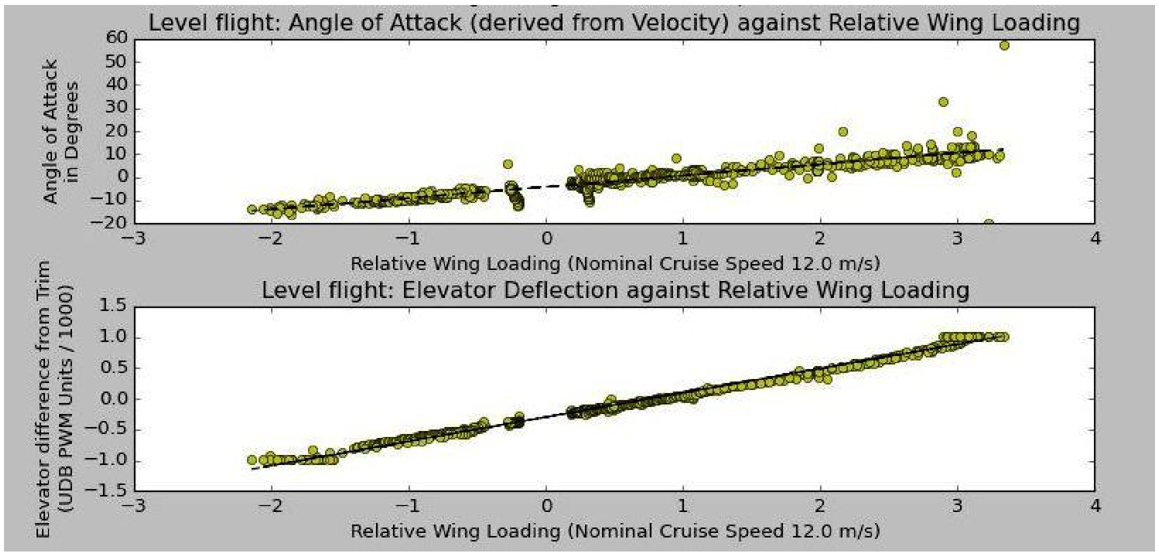

Figure 3 - Angle of attack and elevator trim as a function of relative wing loading

The angle of attack and elevator trim models will improve the performance of the controls during inverted flight. However, there is one other issue to address: GPS performance during inverted flight. If you mount your GPS for best performance during normal flight, it will not work very well during inverted flight, it is likely to stop working all together. One solution is to mount your GPS vertically, for example as shown in Figure 4.

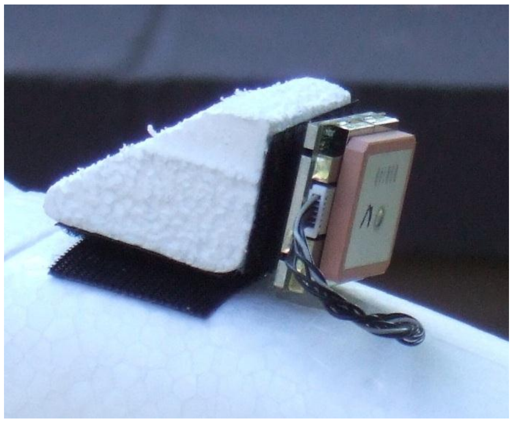
Figure 4 - Vertically mounted EM506 GPS

Flight testing with the EM506 shows that the use of a vertical mount yields inverted performance comparable to performance in normal orientation. Although sensitivity is not quite as good horizontal mounting and normal flight, the sensitivity of the EM506 is adequate to provide good overall performance. For example, under conditions that enabled a horizontal EM506 to access 10 or 11 satellites, a vertical EM506 was able to access 8 or 9 satellites, with the same reported value of horizontal dilution of precision.