Flight simulator mathematics

by Ken Tucker

Behind the fancy graphics of flight simulator programs are complex algorithms controlling the parameters of flight. While complex, an examination of this subject should be of interest to all. This article, while trying to maintain a layman's approach, includes enough mathematical detail to allow a programmer with some knowledge of vectors and trigonometry to experiment with flight simulation programming.

A flight simulator program may be considered as two separate sections: one section for parameter calculation and another for visual representation. The aerodynamic calculations software determines the aircraft's position, speed, direction and orientation. The cockpit view software then uses this data for determining the appearance of the environment, and the instrumentation read-outs.

Flight simulation

The primary forces operating on an aircraft are *gravity*, *lift*, *thrust* and *drag*. By adding the magnitudes and directions of these forces, a single net force is obtained. We can simplify the math involved by setting the weight of the aircraft to one, and by considering force and acceleration to be the same thing. The net acceleration is the vector sum:

```
A = G+L+T+D
```

This could be accomplished in a program through the use of arrays, storing the X-axis component of A in A(1) and using A(2) and A(3) for the Y and Z axes (we will take the three axes as respectively corresponding to longitude, latitude and altitude).

The other 4 vectors, each with 3 components, can similarly be represented, and the sum would become,

```
for n=1 to 3
A(n)=G(n)+T(n)+D(n)
next n
```

As an example, consider an aircraft while it is motionless on the runway. The force of gravity downward is equal in magnitude but opposite in direction to the lift achieved by the wheels. The state of the airplane can be described as:

```
G = -1k
L = 1k
T = D = 0
A = 0
```

In our programmers' notation this is represented by:

```
G(3)=-1: L(3)=1
```

All other array elements are equal to 0.

With the acceleration of the aircraft known, it is very simple to determine the appropriate velocity (speed and direction) and position with a computer. The process by which this is accomplished is known as integration. When a force (acceleration) acts for a period of time, the velocity changes and, while the aircraft is moving, the position is changed:

```
V = V+At
P = P+Vt
```

where V is the velocity, t is the time increment, and P is the position.

In a computer program the time increment (t) can be of any duration, but is usually set so that the program runs in real time. This is accomplished by determining how long it takes the computer to complete one loop through the equations and setting t to that duration.

Wing forces

A wing produces lift through the movement of air, mainly over the top surface of the wing, resulting in upward suction. Actual forces are usually determined by experimental means in a wind tunnel. Recently, computers have been able to achieve theoretically calculated results that approximate experimentally derived results very closely.

Lift actually depends on such broad considerations as wing shape, size and texture, humidity and air density, but air speed and angle of attack are the most important. The greater the angle of attack, the bigger the 'bite' the wing takes from the air, displacing more air downward and producing more lift. All other conditions being equal, if one doubles the air speed the lift is quadrupled. Since the characteristics of the wing itself are fixed for any given aircraft we choose to simulate, we can combine the characteristics for that aircraft into a wing constant W. Then, where V is the airspeed (the magnitude of the velocity vector), and α is the angle of attack, we can derive w, the force on the wing, with:

```
w = V²Wsinα
```

provided that $-10° < \alpha < 18°$. If the wing takes too large a 'bite' (that is, the angle of attack falls outside the specified range), a 'stall' occurs and w becomes zero — there is no lift on the wing. The angle of attack is adjusted by the pilot moving the control column back and forth. This action changes the elevator flap which, in turn, points the aircraft's nose up or down in relation to the wind.

The force acting on the wing is mostly upward (lift), but a comparatively small backward force is also present. This is termed *drag*. The lift occurs in an upward direction relative to the aircraft, while the drag occurs in the direction opposite to that of the aircraft's movement. If u is the unit vector upwards relative to the aircraft's motion, and v is the unit velocity vector (V/V), then the lift L and the drag D can be determined with:

```
L = w cosα u
D = -w sinα v
```

Determining drag

The result for drag in the last equation is only approximate. The fuselage and stabilizer fins, among other factors, create further drag, known as *parasitic drag*. Attempts to eliminate parasitic drag resulted in the famous flying wing experiments. A more accurate value for total drag can be obtained by incorporating a co-efficent of drag, c_d in the equation. This coefficient is determined from the real aircraft's maximum level speed at maximum thrust:

```
D = (-wsinα + c_d V²)v
```

Thrust

Thrust on an aircraft results from either a propeller or a reaction engine such as a turbo jet. A jet engine's thrust magnitude is set by the pilot. The thrust direction is the direction the aircraft is pointed in. We'll assume here that the direction of flight is the direction the aircraft is pointed in. Normally this is true, but there are exceptions. In the famous aerobatic manoeuvre known as the hammer-head stall, the airplane is pointed up, while travelling down. Another famous exception is the Harrier jump jet, whose thrust direction is variable. Ignoring these exceptions, thrust becomes simply:


```
T = Tv
```

As an illustration, remember that the aircraft's weight is one. The actual weight of an F-104A Starfighter is about 20,000 pounds, and it has a maximum thrust of 13,000 pounds. The maximum T in this case is 13,000/20,000 or 0.65. A most familiar unit expressing this is 'g' force; at full thrust, this force is 0.65 g.

Determining the thrust of a propeller driven aircraft is more complex. The pilot is setting the horse power (Hp) of a reciprocating engine from which thrust magnitude must be derived by the formula:

```
T = Hp/V
```

From this equation it's easy to see why a propeller-driven aircraft performs better at low speeds. The lower the speed, the greater the thrust for the same power. Compare the ease with which a helicopter takes off vertically to a Harrier, which requires relatively huge engines to generate more than 1g of upward thrust for vertical take-off. In reality, of course, if the speed is zero (V = 0), thrust is not infinite, as our equation would suggest, since the propeller is still moving. One must refer to experimental data to determine thrust at full power with brakes on. A workable simulation is achieved with:

```
T = Hp/(Vp+V)
```

where Vp is prop speed under these conditions.

Aircraft orientation

Pitch, yaw and roll is not a new sport. It is the dance aircraft wiggle to when flying. Pitch is the up and down motion of the nose, while yaw is the left to right motion of the nose. Roll refers to rotation around the direction of motion. Of these three movements, only roll is directly controlled by the pilot. (Strictly speaking, minor changes to yaw and pitch *can* be introduced directly with the rudder and elevator respectively, but we need not take this into account). This allows considerable simplification with only minor penalty. Letting V_g be ground speed, and using θ for the pitch and Φ the heading (which, because of our simplification, is the yaw angle), we find:

```
Vg = √(Vx² + Vy²)
θ = tan⁻¹(Vz/Vg)
Φ = tan⁻¹(Vx/Vy)
```

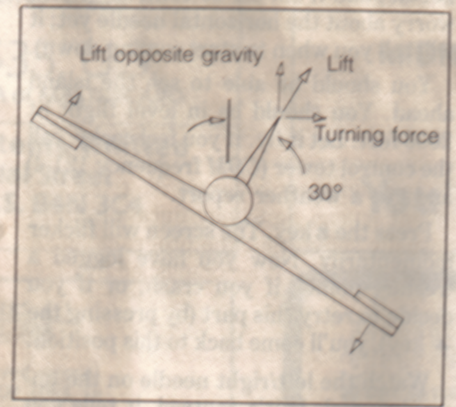
where V_x, V_y and V_z are the aircraft's speed along the three axes.

Most real airplanes use a steering wheel. Turning the wheel left or right causes the aircraft to roll left or right.

The specific response of the aircraft to this control input is highly individual. Transportation aircraft are designed to be very stable with a comparatively large resistance to barrel rolling. Fighters however, need maximum manouverability, and are often designed unstably to this end. The result: a barrel roll at the flick of a wrist.

Turning

An aircraft flying straight and level has a lift force of equal magnitude but of opposite direction to the force of gravity. As mentioned earlier, the lift force is upwards relative to the aircraft. If the aircraft is banked (rolled 30 degrees, say) the up vector will now be pointed to the side of the aircraft, resulting in a turn (see diagram). Notice that the 'lift opposite gravity' decreases as the aircraft is banked. In real aircraft, as in



simulators, the pilot will increase the total lift by increasing the angle of attack. This keeps the 'lift opposite gravity' force constant so that no change in attitude will occur while turning.

Those readers who are mathematical masochists may enjoy confirming how our little unit 'up' vector expands to:

```
u1=-sinΦsinθcosT
    +cosΦsinT
u2=cosΦsinθcosT
    -sinΦsinT
u3=cosθcosT
```

where T is the roll angle.

Conclusion

A proper treatise on flight simulators can fill a text book. Here we have examined a 'first approximation' simulator that is realistic 95 per cent of the time. The next step (second order approximation) requires consideration of the aircraft's moments of rotation, fin stabilizing geometry and so on, to enable the simulation of spins, stalls and minor lags resulting from the difference in the direction of flight and the direction the aircraft is pointed. As the software grows more complex, the loop time increases, with the projected result becoming increasingly accurate but too slow for comfortable use. All real-time flight simulators must make a trade-off between speed and precision; the product of these two factors is a measure of the brute hardware power of the machine running the program.

When one considers the additional computer time required for the view simulator, instrumentation and possible statistical updates, the advantages of parallel processing become clear. Flight simulators have a natural affinity for parallel systems of two or three processors. One complex task, the flight simulation, passes only six numbers to the next complex task, the view simulator. With three axes of position and three angles of orientation, the view of an environment can be calculated. Without doubt, as hardware capability expands, an entirely new class of simulators will become practical and accessible to the home user for both entertainment and education. □