Flight Models in Flight Games

How *Ace Combat 4* and *Sky Odyssey* juggle playability and realism

Bernardo Malfitano
STS 145 final paper
Winter 2003

Introduction

Flight games are extremely varied, and one of the ways in which they vary is the realism of their flight control models. The easy-to-learn but not realistic control models of space shooters and console "fighte jet" games make for very different playing experiences than the detailed, thoughtful, realistic models in flight simulators. There is a spectrum of flight control models ranging from simplistic to realistic, offering different advantages and disadvantages. Here, two flight models will be analyzed, as will their consequences on gameplay, so that the advantages and disadvantages of different choices during the creation of flight models can be understood.

A history of flight games and their different varieties and goals, followed by a summary of real flight control dynamics, will set the stage for a thorough analysis of the flight control models in Namco's *Ace Combat 4* and Activision's *Sky Odyssey*. The characteristics of these models will then be compared to the goals and gameplay of the games, so that the reasons for the characteristics of the flight models may be better understood.

A brief history of flight games

Flight games and flight simulators have a long history to them. Since the beginnings of videogaming, most games portray activities that gamers will probably never experience – driving very fast, very expensive cars at very illegal speeds, competing in national or international-level sports, being a soldier in combat, slaying dragons, battling alien races, traveling through space. Flying combat aircraft is a job that requires training, discipline, skill and courage, but that implicitly carries enormous power, navigational and destructive. Fighter and attack aircraft are exciting because they are not built for efficiency – they are fast and aerobatic, high-performance vehicles with unparalleled powers of destruction. This is why some of the earliest games, like *Chopper Command* or *River Raid* and its sequel, had they player controlling crude representations of fighter and attack jets.

As games became more sophisticated, the rewards and excitement of commercial and general aviation could be reproduced to a satisfactory level, and flight simulators were born. These flight games were not about speed, explosions and tight turns. They were about the science, freedom, beauty and adventure of flying.

In military flight games, the rewards for the player were the visuals, the simulated battle, the explosions, the feeling of power in controlling a fast, armed airplane. Therefore, controls were simplified, gameplay was often projected/reduced to one or two dimensions, and making the virtual airplane move was hardly more complicated than moving a cursor on a computer. The challenge – like the reward – was in the shooting, not the flying.

However, much of the reward of real flying comes from mastering the controls of the aircraft so as to make complicated, difficult maneuvers more intuitive, so as to exploit the unique freedom that an aircraft provides. Therefore, these non-military flight games had more complicated controls that worked more and more similarly to the controls of real aircraft. Nintendo's *Pilotwings* is an early example, and even today is quite unique because it was made for a console rather than for the PC.

As games became even *more* sophisticated, both graphics- and gameplay-wise, gamers came to expect more from games. Military flight games had to be as graphically complicated and immersive as non-military flight sims. So as to exploit the 3D nature of these new immersive world, the controls of these military planes had to allow for navigation in 3D. This is when they, too, started to mimic the controls of real aircraft. Meanwhile, non-military flight sims became so graphically sophisticated that the enjoyment from the part of the player came as much from the "look" of the environment around their airplane (and of the airplane itself) as from the satisfaction and challenge of mastering the controls. So some flight sims aimed to be less realistic so as to be more enjoyable. Less investment from the player was required to learn the controls.

Soon, flight games made a somewhat continuous spectrum from the not-at-all realistic shooter to the as-realistic-as-possible simulator. At the "shooter" end were space combat simulators and very few fighter jet games. (It is interesting to note that the controls of most 3D space shooters are loosely based on aircraft controls – just about the most intuitive and freeing way to move in three dimensions). Most of these games are console games. At the "simulator" end were flight simulators like Microsoft's *Flight Simulator*, and most were made for home computers.

Flight games released for the consoles over the past five or six years have featured progressively more and more realistic controls. Nintendo's *Pilotwings 64*'s gyrocopter had very detailed flight models. The Playstation release of *Top Gun* featured a Tomcat that reacted realistically to control-surface damage. The jets in Namco's *Ace Combat 3* could reach high angles of attack, the first time a console game featured fighters that were not infinitely stable. They also performed significantly more sluggishly at high speeds and altitudes. Dreamcast games such as Konami's *Air Force Delta* and Crave's *Aerowings* first applied these more complicated dynamics to the current-generation consoles. *Ace Combat 4* and *Sky Odyssey* were two early releases for the Playstation 2, and reached a never-before seen level of realism in console flying. However, they still made many compromises in their flight models in order to be console-friendly.

What are the trade-offs between realism and ease of gameplay? For different kinds of games, what are the advantages and disadvantages of a simplified or a realistic flight control model? The answer to these questions should reveal much about why flight games are made the way they are, and about the values of different places in the realism spectrum.

Flying a plane

An analysis of simulated flight controls cannot be meaningful without a short description of the real controls they mimic. This section will describe approximately how a real airplane is controlled.

The three variables over which a pilot has constant, analog, real-time control are the roll axis, the pitch axis, the yaw axis, and the thrust of the airplane.

(Other control factors are rarely used and not subject to analog control, such as landing gear settings or flap settings – i.e. the wing's low-speed lift coefficient. Other important aspects of flying – such as navigation and communication – are relatively inconsequential to the flight controls.)

The pitch axis runs horizontally in the left-right direction, usually from wingtip to wingtip. Movement in the Pitch axis is achieved by pulling or pushing the control stick/column, causing the nose of the airplane to rise or fall – in other words, causing the airplane to point up or down.

The roll axis runs horizontally in the back-forward direction, usually from nose to tail, through the pilot. Movement in the roll axis is achieved by tilting the control stick/column to the left or right, causing opposite wings to rise or fall – in other words, causing the airplane to bank.

The yaw axis runs vertically through the middle of the airplane. Movement in the yaw axis is achieved by pushing the left or right rudder pedals, causing the nose of the airplane to move to the left or right, much like a car making a turn.

The thrust put out by the engine is either controlled by a single lever, in more automated planes, or by the adjustments of several variables in the engine operation, such as fuel-air mix, the rate of fuel injection, and the orientation of the propeller blades. Thrust is a force that, except in very few military airplanes, pushes the plane in the direction of the nose (forwards).

From this simple description, a simplistic flight model could be made, and it would be similar to the flight models of most early simulators. Pressing “Right” or “Left” would cause the airplane to roll at a certain rate. Pushing “Up” or “Down” would cause the plane to pitch down or up, at a certain rate. Two keys would be assigned to the rudder pedals, and would turn the airplane sideways like a car, at a certain rate. The throttle control would adjust the thrust, making the plane go faster or slower – different throttle settings correspond to different terminal speeds. The airplane always goes in the direction its nose points.

Such a control system would not take long to learn, and would not be a challenge in itself. However, it would not be realistic.

This is because an airplane does not always fly in the direction it points. At slower speeds, the wings need to be tilted upwards more, in order to move enough air enough to generate the lift needed to keep the plane in the air, than at faster speeds. This means that a plane flying more slowly will have its nose pointed more upwards relative to the direction of flight, and that a plane flying more quickly will have its nose pointed more downwards relative to the direction of flight. The angle between the nose (or the imaginary line of the roll axis) and the direction of flight is the *angle of attack*. It changes with speed, altitude and load. Some games take it into consideration, some do not. Until recently, in most console games an airplane always flew in the direction it pointed.

When the angle of attack becomes too great in a real airplane, the air flowing over the back of the wing stalls, and the airplane loses lift (in turn increasing the angle of attack, making the airflow over the control surfaces very different). This effect, where a critical angle of attack causes sudden differences in flight direction and control, is often not simulated in games, at least not well.

There is also a lateral angle of attack, called the angle of sideslip, which changes when the plane turns. When the plane yaws, it is facing off to the side of where it was facing before, but unlike a car it does not move in that direction immediately. It takes a short amount of time for the new component of the thrust, and the difference in pressure from one side of the fuselage to the other, to make the plane move that way. Similarly, when a plane is banking, it creeps to the side as well as turning itself.

The angle of attack and sideslip present a new factor: stability. It is crucial for the control of aircraft, and is often the defining factor on whether a plane in a game flies like a real airplane.

The tail surfaces of an airplane are there so that the plane never points too far from the actual direction of flight. An angle of attack or sideslip would push the tail surfaces behind the airplane so that it faces the direction where it flies. For this reason, a plane maintaining a low speed needs constant pitch-up input to fight the natural tendency of the angle of attack to return to zero. When a plane banks and moves sideways, the tail turns the airplane by being pushed behind it by the sideways-moving air.

Many early flight games featured infinitely stable airplanes: the plane always moves in the direction it points, banks do not cause turns (the player had to pull up on the stick to pull the nose up into a turn), etc. Even some relatively recent flight games like *Independence Day* work like this.

Many other flight games feature infinitely unstable airplanes: a loss of speed during level flight would cause an increase in angle of attack without the nose pitching downwards. The plane would soon be falling onto its belly, or maybe even partially backwards it if started out in a climb, without its orientation in space changing. These virtual airplanes flew very differently from the ones that were at all stable, because it is hard for the "pilot" (who is not in a real aircraft and cannot feel the virtual aircraft change the direction of flight) to tell the airplane is no longer flying where it is pointing.

There are other factors that affect control which some games include and some do not. An airplane changes its orientation by moving control surfaces on the wing and tail. These surfaces deflect air and so are pushed by the air, thus tilting the airplane in that direction. Flight at slow speed or high altitude means less air passes the control surface per unit time, so the rates of roll, pitch and yaw are slower. Rates of turn are also slow at supersonic speeds, where shockwaves disturb the air flowing over the airplane.

So a flight control model in a game or simulator can be more or less accurate in many different ways. In different aspects, *Sky Odyssey* and *Ace Combat 4* lie at different places in the spectrum between simplified and realistic. These aspects and their consequences will now be analyzed.

Test-flying *Ace Combat 4* and *Sky Odyssey*

Although *Sky Odyssey* features many fantastic and fictional aircraft, it also features many real-life aircraft, and those are the ones this analysis will regard. *Ace Combat 4* also features one fictitious jet, but it does not behave much differently from the high-end Sukhoi Flankers in the game.

As the first step of any flight is the takeoff, the analysis will start with a comparison of low-speed handling characteristics of the flight models.

The aircraft in *Ace Combat 4* are realistically stable. The angle of attack increases as the speed decreases in a realistic way, requiring the player to pull the joystick back further the slower they fly. Below a certain speed – different for each aircraft – the nose-down tendency generated by the stable reaction to the high angle of attack cannot be overcome, and the airplane noses down (and thus gains speed and reduces the angle of attack) if the thrust is not increased. The nose-down tendency increases abruptly below a certain speed, where the angle of attack rises quickly. This is a very good simulation of a stall, and makes for very realistic takeoffs, landings, and slow flight. Roll, pitch and yaw control is realistically more sluggish at low speeds.

The aircraft in *Sky Odyssey* are extremely unstable. Upon landing, or simply slowing down to near landing speeds, the angle of attack initially rises realistically, but the nose does not drop. The stall does not come quickly; the angle of attack simple increases as speed decreases until it is an absurd 90 degrees. At this point the player must increase thrust in order to gain speed, because 3-axis control at high angles of attack is sluggish to the point of being useless. Paradoxically, control is still very responsive at takeoff and landing speeds. Its effectiveness drops significantly and quickly past a certain very high angle of attack.

Despite the poor modeling, especially when it comes to the zero stability throughout all speeds and angles of attack, it must be noted that *Sky Odyssey* allows you to operate the landing gear, an extremely rare feature in console games.

Altitude effects in both games are very poorly simulated. In *Ace Combat 4*, reaching a certain altitude (different for each jet, but in the range of 40000 feet) causes the jet to instantly stall. The angle of attack jumps from near-zero to near-90, and the nose drops towards the ground. Speed also falls unrealistically fast. Although a wing will stall more and more easily in the thinner and thinner air of progressively higher altitudes, this is not what happens in this game. The stall speed for the jet is the same for all altitudes, until at the “ceiling” it is *any* speed. The absurdity of this model is best shown if the player flies a few hundred feet under the ceiling at max afterburner for some time to reach the jet’s maximum speed, then turns upwards into a vertical climb. Upon reaching the ceiling, the plane snaps downwards, turning one hundred eighty degrees in less than a second.

Other effects of altitude change, such as loss of control-surface effectiveness or changes in maximum speed, are not modeled in the game. The range of speeds attainable by each jet is the same at sea-level than just under the ceiling. Pitch, roll and yaw control changes realistically with speed (poor at low speeds, very poor at very high speeds, optimum at high subsonic speeds), but not with altitude.

(It is interesting to note that in *Ace Combat 3*, a Playstation 1 game, high-flying jets such as the SR-71 became significantly more sluggish at altitude, and that in *Ace Combat 2* and *3*, jets’ stall speeds increased at high altitudes even before the mandatory high-speed stall at the ceiling).

Sky Odyssey does an even poorer job of simulating a ceiling. The airplane is suddenly subjected to a very strong crosswind when reaching a certain altitude. The wind causes a high angle of sideslip (lateral angle of attack). As with high angles of attack during slow-speed flight, the control surfaces become ineffective and the plane simply falls onto its belly until the crosswind is over.

But the ceiling is very hard to reach except with the game’s highest-performance airplanes. Slower airplanes – including all the planes available before beating the game for the first time – gradually lose 3-axis control and engine responsiveness after a cetain high altitude. The angle of attack also increases. Except for the instability and the lack of a well-defined stall as discussed earlier, these changes with altitude are realistic. In real life they happen over greater altitude changes than in the game – the game renders the terrain and weather only when very close to the player, so if the players are allowed to fly realistically high, they would lose sight of the ground – but the changes in the game are gradual enough to be recognized and dealt with by the player. One very challenging level in the game has the player flying over a mountain pass. The engine does not produce much thrust, the angle of attack is high, control is sluggish, and only by dumping fuel – another unique ability in this game – can the plane be light enough so that these small forces maneuver it at high altitude.

Effects in gameplay

What does it mean for the player that these simulated airplanes behave in these ways?

Stability and angle-of-attack effects are beneficial in both games. The stability in the airplanes in *Ace Combat 4* means the plane is flying in the direction it is pointing. In a combat flight game, this is very important, as flying *and* pointing towards the player's targets allows them to fly intuitively and "aim" their airplanes at their targets.

But in a game like *Sky Odyssey*, where the navigation of the aircraft and the performance of aerial stunts is more important, instability and a poorly-modeled absent stall allow for controlled slow-speed flight and precise navigation through tunnels and around obstacles, and landings made in small spaces. Thus, the "mistakes" in the flight model in Sky Odyssey actually make for easier gameplay. In fact, an external camera angle available in the game looks to the direction the airplane is flying – so, for example, during high-angle-of-attack flight, the camera looks at the top/back of the airplane. This allows the player to know the angle of attack in a very precise and useful way, and to be aware of changes to it in a way that a cockpit camera or behind-the-plane camera would not reveal.

In a combat flight game, a player will often want to fly as high as possible, in order to avoid the enemy while on the way to a target, as ground-based enemies cannot reach the airplane and airborne enemies need time to climb. While the presence of a ceiling is initially annoying and frustrating when it comes to wanting to fly high, it does make it easier to stay at the optimal flight altitude. Questions such as which altitude allows for the fastest max speed, or the lowest fuel consumption, become trivial when the flight ceiling is fixed and other flight properties do not change with altitude. This is one example where inconsistent physics helps the game in a symbolic way, as in the classic example in Steven Pool's *Trigger Happy* of Lara Croft's inability to destroy a locked door with a grenade launcher. This is also the role of the ceiling in *Sky Odyssey*: If the ceiling were not unrealistically low, the player would simply fly over most obstacles.

In a way, the instability in the aircraft in *Sky Odyssey* make landing (often the most challenging part of a flight game – in the few which care to include landings – and the most challenging part of learning to fly a real airplane) easier. Even at high angles of attack and low speeds, the player can know that the attitude and orientation of the airplane will not change unless made to change by control inputs. The nose will face the horizon unless pitch-change inputs are made, and the airplane will not bank to one side at high angles of attack (as some airplanes do), so the player needs only to control the speed of the airplane during landing, not the orientation. The airplane will land upright. Since landing is crucial to the completion of each flight in the game, and because it comes in the end (after all the other work has been done to get to the end), it is easy to see why it is worthwhile making landing easy – otherwise it would easily be the most difficult and frustrating part of the game.

Low-speed handling in *Ace Combat 4* must necessarily be tricky, because the easiest way to win a dogfight or to precisely drop a bomb is to fly very slowly. In a dogfight, the opponent would fly past the player and be right in the player’s gunsights or in the perfect place for a Sidewinder target. In a bombing run, the more time the player has to aim, and the slower the target moves relative to the player, the better.