

Technophobia

Not an irrational fear; it's a survival mechanism

BY JAMES ALBRIGHT james@code7700.com

technophobe

['tekne fob']

NOUN

A person who fears, dislikes or avoids new technology.

Yes, I am a technophobe in the strictest sense of the definition, with an emphasis on the word “new.” I don’t trust anything that hasn’t been proven.

Of course, it wasn’t always that way. I embraced computers early on. As an engineering student in 1974, my interaction with computers was via punch cards, and you counted lines of codes by the thickness of your card deck.

In 1983, I wrote a program for my U.S. Air Force Boeing 707 squadron to automate the chore of computing navigation and fuel logs and I thought that was great until we discovered an error in one of my lines of code. My mistake impacted all four of our squadron’s airplanes. In 2004, another software error in an air traffic control computer impacted 800 airplanes.

As aircraft become more and more dependent on computers, the result of a “software glitch” becomes harder and harder to predict. If an airplane is completely airworthy when flying on one side of a line of longitude, how could an imaginary line in space prevent it from flying according to specs on the other? Or how about another airplane whose electrical system now has a calendar-based limitation? Examining a few examples where these glitches were thought to be unpredictable can help us better prepare ourselves for what we cannot possibly foresee.

Going Digital for the First Time

In 1982, I was a copilot flying U.S. Air Force Boeing 707s (EC-135Js) with only one thing in the cockpit that could be remotely called a computer, and that was the flight director. A mission always started the day before with the

navigator pulling out a paper chart, a plotter and dividers. After a few hours he would have a navigation log. Then I would take that log and a book of charts and add fuel computations. I would hand all of that to the aircraft commander, who would pick up the phone and order a fuel load for the next day’s mission. This ritual took about 4 hr. All that changed the following year when the squadron got its first desktop computer, an IBM PC 5150.

The original IBM PC was an 8-bit computer. (More on that later.) Using

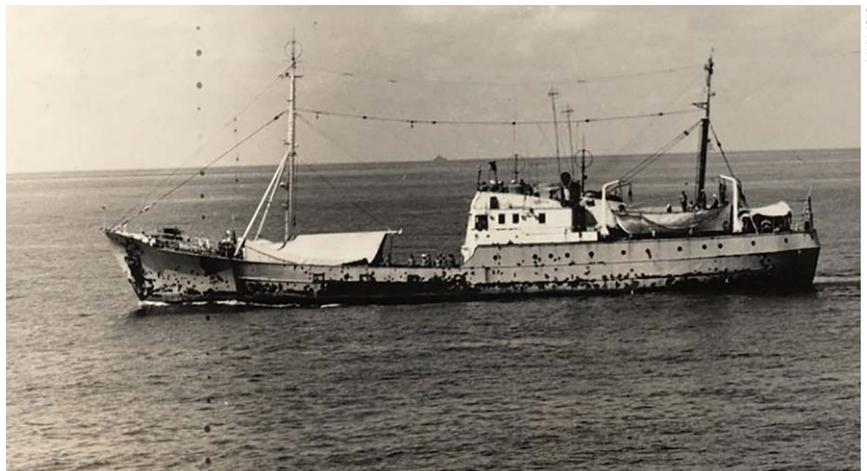
what I had learned in college by programming mainframe computers, I wrote a program to automate the chore of preparing navigation and fuel logs. After that, our squadron started churning out navigation logs in minutes instead of hours.

Before I got the program running, I upgraded to aircraft commander and was assigned to fly a mission our squadron only got once every few years and it required some unusual flight planning. We were required to fly just a couple of hundred feet above the water

and head right at Russian spy ships decked out to look like fishing trawlers. These ships became known as AGIs, since the U.S. Navy classified them as “Auxiliary, General Intelligence.” We were told they were armed and that it was best for us to fly as fast as possible, as low as possible. So, that’s what we did. When I wrote the fuel-planning software, I made sure it



IBM PC 5150, circa 1982



Soviet AGI, 1970

FMA



Air Route Traffic Controllers

out the airborne traffic. There were no accidents, but at least 10 aircraft had what is euphemistically called a “loss of separation.” News reports from the time said: “Exactly what went wrong was not immediately determined.”

There isn’t much official about the incident, but it seems it was due to a timekeeping error with the center’s computers, which tracked the passage of time by starting with the number 4,294,967,295 and counting down once a millisecond. That number, you may recall, is what 32 bits comes to in Base 10.

Which meant that it would take 49 days, 17 hr., 2 min. and 47.295 sec. to reach zero. Can time go negative? Perhaps, but apparently not in this computer. Usually, the machine would be restarted before that happened, and the countdown would begin again from 4,294,967,295. I’ve read that some people were aware of the potential issue, so it was policy to restart the system at least every 30 days. But in this particular cycle, the computer ran for 50 days without a restart, hit zero, and shut itself down.

If you ever had a 32-bit computer that you allowed to go into “sleep mode” regularly rather than have to wait for the full boot-up process the next day, you may have noticed the computer crashed on you now and then. Perhaps it was 49 days since its

last crash. Now you know why.

The fix would be to rewrite the code so that instead of shutting down, it simply restarted the count on its own. Modern computers these days are using 64-bit processors, which, if instructed to count backward from 2^{64} in milliseconds, would end up at zero in 584.9 million years. So, we can worry about this later.

Technophobe Technique. In the days following Sept. 11, 2001, a NOTAM was issued in the U.S. requiring crews to

monitor emergency frequency 121.5 MHz, but that mandate was allowed to expire. In some parts of the world there are backup air traffic control frequencies. Well, guess what? There is a backup frequency here, too, and you should make use of it: 121.5 MHz.

The F-22 and the International Date Line

If you’ve ever flown an airplane that came right out of the factory you will probably have experienced what the U.S. Navy calls the shakedown cruise, an initial outing to shake out all the glitches. But in an airplane, the shakedown lasts several months. If you are flying a brand-new aircraft type, it could last years.

At the time of its introduction, the Lockheed Martin F-22 Raptor was the pride and joy of the U.S. Air Force. At \$360 million per aircraft it wasn’t cheap, but it represented the cutting edge in technology. In February 2007, a flight of six Raptors was scheduled to make their first deployment from Hawaii to Japan, on the wing of a refueling tanker.

All was going smoothly until they crossed the International Date Line, when the computers on all six fighters stopped working. The pilots were left without navigation, communications or fuel management. Attempts to restart the systems proved unsuccessful. Fortunately, their fly-by-wire systems were still working, they were able to visually signal the tanker to turn around, and they had enough fuel to make it



ROB SHERK

F-22 Raptor, May 17, 2008

back to Hawaii. The cause was said to be a “computer glitch in the million lines of code.” The Air Force never confirmed what went wrong, only that it was fixed within 48 hr.

Technophobe Technique. Be mindful of an aircraft’s limitations and history. If it hasn’t been done before, realize you are a de facto test pilot and need to develop an exit strategy for those periods where you are quite literally flying into the unknown. A better plan would be to leave the experimental flying to the experimental test pilots.

The Boeing 787 and Dreams of Simpler Systems

In 2015, the FAA issued an Airworthiness Directive that, at first reading, seemed more of a prank than anything else. When have you ever heard of an aircraft electrical system with a calendar-based limitation? Well, here you go:

“This AD was prompted by the determination that a Model 787 airplane that has been powered continuously for 248 days can lose all alternating current (AC) electrical power due to

care about the 248th day any more than the 247th? There were a number of “tech blogs” at the time that hypothesized there was a timer in the GCU that counted in 100ths of a second and that $100(2^{31}) = 214,748,364,800$ sec., which when divided by the number of seconds in a day equals 248 days and therefore the computer in the GCU couldn’t multiply beyond 2^{31} for some reason.

But that makes no sense because 2^{31} means nothing in the computer world. But what if you dedicate the first bit of that 32-bit data register to a plus or minus sign? That would leave you with enough bits left over for 248 days, 13 hr., 13 min. and 56.47 sec. counting in 100ths of a second. The fix has been to require the airplane to be rebooted more often.

Technophobe Technique. As the aircraft I’ve flown have grown with complexity so has the “cold boot” process. I’ve noticed that if the boot-up process doesn’t go the same way it usually does, chances are something isn’t going to work correctly later in the flight. I think you should carefully watch and record what a normal boot-up looks like.

A cellphone video of the cockpit during a normal boot gives you a quick reference for when the boot-up just doesn’t seem right. It may save you time in the long run to reboot as soon as you see the cold boot isn’t progressing normally. After 10 years with the Gulfstream G450 we got pretty good at predicting this. If you are going to have to reboot anyway, it is better to get it over with early rather than have to do almost

the entire preflight twice. We are now flying a new aircraft, and we are still learning. And still recording.

What Is a Line of Code?

You often hear manufacturers beg forgiveness for technical glitches by saying it was impossible to predict because of the millions of lines of code in the aircraft’s computer systems.

What does that mean? Consider a student computer programmer’s first program, the so-called “Hello World Program” written in a computer language that is so basic, it is called BASIC (Beginner’s All-purpose Symbolic Instruction Code).

10 PRINT “Hello World!”

That’s one line of code. Now let’s make it just a little more complicated:

10 PRINT “Hello World!”

20 GOTO 10

Now you have two lines of code and there is an error. The error is called an “endless loop” because the program will just keep printing “Hello World!” over and over without end.

It was an easy error to spot. The F-22 Raptor’s error was said to be somewhere amongst 2 million lines of code. The error in my flight-planning software for the Boeing 707 was buried in just over 5,000 lines of code and the only way I found the glitch was to have someone actually attempt to fly the specific condition.

An Example of a Serious Line of Code Error

The Gulfstream GV was a game-changer in business aviation, offering unparalleled speed, range and efficiency. In 2004, the model was updated with the Gulfstream G550, the same basic airplane with a brand-new avionics package. The glass cockpit with four large display units (DUs) was revolutionary and has gone on to be the cornerstone of some very good cockpits in the Gulfstream stable. The airplane got its type certificate on Aug. 14, 2003.

The next year, a line of computer code caused all four DUs to go blank on at least one crew while in flight. Before the code could be rewritten, Gulfstream issued Maintenance and Operations Letter G550-MOL-2004-0020 recommending the following procedure if the DUs failed to recover within 90 sec.:

If DUs DO NOT recover, continue checklist with Step 7:

(7) Advise passengers that normal power will be off.

(8) Left and right generators (and APU generator, if in use) . . . OFF.

(9) Wait 30 seconds.

Gulfstream issued a software

“NEW YORK AIR” PHOTO



Boeing 787-9 at the 2019 Paris Air Show

the generator control units (GCUs) simultaneously going into failsafe mode. This condition is caused by a software counter internal to the GCUs that will overflow after 248 days of continuous power. We are issuing this AD to prevent loss of all AC electrical power, which could result in loss of control of the airplane.”

I read that AD from cover to cover for signs it was a joke. Why does a GCU

MATT BIRCH, HTTP://WWW.VISUALAPPROACHIMAGES.COM



Gulfstream G550 on approach to Henderson Executive Airport (KHND), October 19, 2019

update (Aircraft Service Change 902) that fixed the broken code in December 2004, but the FAA followed up the next year with an Airworthiness Directive that said noise interference in the avionics standard communications bus would interfere with the display recovery.

Technophobe Technique. Can you imagine flying across the North Atlantic at night with nothing more than a standby attitude indicator? For around \$600 you can buy a portable WAAS GPS equipped with ADS-B In and an integral backup attitude system that will link with an iPad. This setup could be a lifesaver the next time ATC shuts down, or you discover a broken line of software code. That is cheap insurance for a multi-million-dollar aircraft.

The avionics suite in the G550 is a work of art and I can imagine there must be millions of lines of code behind it. Is it surprising that a line of code made the wrong decision reacting to noise interference? I don't know, but I can certainly sympathize with the software engineers. What can we, as pilots, do about this? Let's look at one more example.

Example of a Nuisance Line of Code Error

We took delivery of our Gulfstream GVII-G500 within a year of the

airplane's certification and we had on board a Gulfstream test pilot who was more than just a run-of-the-mill Entry Into Service (EIS) demo or delivery pilot; he really knew his stuff. The first thing we did was do three takeoffs and landings at Savannah/Hilton Head International Airport, Georgia (KSAV), each with an ILS approach and a quick reloading of the flight plan in the FMS, all handled by our EIS pilot.

We then loaded a flight plan to Laurence G. Hanscom Field Airport, Bedford, Massachusetts (KBED). The FMS initialized the route with a speed of 160 kt. and cruise altitude of 180 ft. Try as we might, we couldn't change that. Our EIS pilot called all the company experts and they all said, "We've never heard that one before." Finally, someone suggested pulling and resetting the Modular Avionics Units circuit breakers. That worked, but it made the cockpit go nuts. I told the EIS pilot, "Let's never do that again."

The pilots at Gulfstream were stumped. But how often did they fly a local pattern and then set up for a Point A to Point B trip? Perhaps never. I reached out to a GVII user's group and discovered that a few operators already found a workaround: Hit the Take Off/Go Around button on the thrust levers to fix things. That worked. After some experimentation, we discovered

the problem only seems to happen after flying an ILS, shutting the engines down but leaving the avionics powered, and then entering or downloading another flight plan. We made several videos of the problem and now the avionics manufacturer is hunting for a line of code to fix for a problem never anticipated.

Technophobe Technique. Manufacturers and training vendors tend to live in sterile bubbles where conditions are largely predictable. There is no way they can anticipate and simulate every possible condition "out there" in the real world. It is in everyone's best interest if those flying a particular aircraft

type connect with the rest of the community and share experiences. In the few months I have been flying our new airplane I've already discovered a few problems that stumped Gulfstream and our training vendor but found answers in a user's group. "They" build the airplane or train us to fly it. But "we" have more relevant, real-world experience. We should share that knowledge.

Technophobia as a Survival Mechanism

Science fiction writer Arthur C. Clarke formulated a series of adages that became known as "Clarke's Three Laws" and of those, the third, applies: "Any sufficiently advanced technology is indistinguishable from magic."

It is tempting to think of those millions of lines of code that go inside our computers as magic. As such, only someone who thinks in ones and zeroes can hope to understand them and the rest of us are at the mercy of the magician. As modern-day aviators we are immersed in technology and the stakes are too high to leave our fates to sorcery. But we can anticipate the unpredictable by playing the "what if?" game and developing techniques to survive.

Technophobia, it would seem, is a valuable tool if used wisely. **BCA**