Automation Resource Management

Managing the **digital crewman**

Pilot Justin Serbent using the Gulfstream G450 guidance panel.

BY **JAMES ALBRIGHT** james@code7700.com

With each new generation of aircraft we are treated to levels of automation unparalleled in performance and reliability. For those of us who first flew with autopilots capable of little more than holding a heading or an altitude, a system that can fly a radius-to-fix approach to a 0.1-nm Required Navigation Performance (RNP) is truly a step into a new universe.

But in all that time we've not progressed from thinking about the electrons-to-neurons interface as anything more than another chapter in cockpit appliances. It is time to consider cockpit automation as another pilot, one capable of routine brilliance but occasional catastrophe. That electronic pilot needs the same crosscheck as its flesh and blood counterparts. Crew Resource Management (CRM) can provide lessons on managing our digital colleague.

It is said CRM came into the mainstream after the crash of a United Air Lines McDonnell Douglas DC-8 four days after Christmas in 1978. Flight 173 departed Denver for a two and a half hour flight to Portland, Oregon, with 181 passengers and a crew of eight. The flight was uneventful until cleared for the visual approach to Portland International Airport's Runway 28. When the crew extended the landing gear, they felt a large jolt. Only the nose gear indicated that it was safely down and locked.

The crew then spent an hour holding while troubleshooting the problem and preparing the cabin for landing. The captain was repeatedly distracted and ignored very gentle suggestions from his first officer and flight engineer that fuel was getting low. The airplane simply ran out of gas and crashed 6 mi. from PDX. Of the 189 persons on board, 12 were killed.

After that, CRM was finally accepted industry-wide. Had the crew been more assertive and had the captain been more receptive, the crash would never have happened. The gear, by the way, was found to be fully extended and locked at the time of impact.

While the concept of CRM was revolutionary in the late 1970s, today we not only accept but also expect crews to constantly check each other's performance and actions during all phases of flight. The other pilot may be highly experienced and have an unparalleled safety record, but mistakes do happen, which is why we have more than just one pilot up front in the first place.

Left unsaid in most CRM discussions is the fact there is a third "pilot" also charged with decision-making and flight tasks: the automation systems. In some aircraft this may be nothing more than a flight director or wing leveler. In others it can be a series of electronic boxes with more computing power than found in the Space Shuttle, capable of conducting the entire flight from takeoff to landing. In the decades since the crash of United 173, we have witnessed scores of aircraft mishaps attributed to troubles with the pilot/automation interface. In many cases the automation suffered from programming errors, the classic garbage-in, garbage-out syndrome. In other cases it was improper system design or the pilot's failure to understand the design. Regardless of the cause, lives could have been saved had the pilots simply kept a keen eye on the automation.

From 'Click, Click' to Startle

One-half of the "click, click" solution

JAMES ALBRIGHT (2)

With the so-called first generation of cockpit automation, we simply executed the command (*i.e.,* pressed a button) and confirmed that the intended action took place. These systems partially integrated autopilot/flight director and autothrottle modes, giving us fairly robust lateral navigation. If something went wrong, we simply disengaged the autopilot and autothrottles and took over.

This "click, click" mentality wasn't usually a big problem, given that the automation was only capable of rudimentary tasks (holding a course, speed and descent rate) and we were usually able to take over quickly. Back then, we were further armed with a healthy mistrust of the electrons; we expected them to fail and we expected to have to take over on a moment's notice.

But today's systems have proven to be extremely reliable, to the point where we tend to think of them as infallible. In fact, most modern aircraft are so fully automated that it would be unthinkable to dispatch without the automation fully operational. But after years of flying without a major avionics glitch, we've developed a tendency to let our guards down. Now, any automation hiccup can result in what has become known as the inadequate "startle response."

A 2012 European Union study titled "Manual Operations for Fourth Generation Airliners" concluded that, "Despite the substantial and proven safety benefits of automation systems in third- and fourth- generation aircraft, evidence indicates that when faced with unexpected and challenging situations, pilots sometimes have difficulties responding to situations which require a rapid transition in their activity from monitors of very reliable systems, to active and authoritative decision-makers exercising manual control of the aircraft."

But the problem goes beyond being surprised by an automation anomaly and the need to take over. Pilots may simply be incapable of flying their airplanes as well as their digital counterpart. Second- and third-generation systems fully integrate lateral and vertical modes and can handle complex tasks with greater speed and precision than even the most capable and proficient human pilot. You need an autopilot to operate in Reduced Vertical Separation Minimum (RVSM) airspace. The once-required instrument pilot skill of "fix-to-fix" navigation fails to meet modern accuracy requirements and may be illegal in some airspace. There are times when having an element of automation failure can mean the trip itself cannot proceed as planned.

The modern response to handling automation problems is to dial back the level of automation to gradually return duties from electrons to neurons — from avionics to humans — to ease the sudden cognitive burden placed on the pilot. This approach has been variously called "level reversion," "stepping back" or "automation shedding." The key point is you turn off the most complex items first, continuing further until the problem is resolved.

Flight Safety Foundation 'Golden Rules'

Automated Aircraft Can Be Flown Like Any Other Aircraft

Aviate (Fly), Navigate, Communicate and Manage – In That Order

Implement Task-Sharing and Backup

Know Your Available Guidance at All Times

Cross-Check the Accuracy of the FMS With Raw Data One Head Up

When Things Do Not Go as Expected, Take Control

Use the Optimum Level of Automation for the Task

The Flight Safety Foundation's (FSF) Approach and Landing Accident Reduction (ALAR) Tool Kit Briefing Note 1.2 offers the following example any time the aircraft does not follow the desired flight path and/or airspeed:

▶ Revert from FMS to selected modes;

▶ Disengage the autopilot and follow flight director guidance;

▶ Disengage the flight director, select the flight path vector (as available) and fly raw data or fly visually (in VMC); and/or,

▶ Disengage the autothrottles and control the thrust manually.

In principle this approach seems to be just what is needed, and yet we continue to see examples of pilots being unable to quickly resolve situations by incrementally lowering automation levels. In fact, when overwhelmed, pilots may be inclined to revert their levels all the way. The FSF's "Golden Rules" encourage a measured approach to level reversion, and conclude with "when things do not go as expected, take control."

A 2010 FSF survey found pilots tend to prefer to hand-fly as much as possible as a way of keeping proficient. But the survey also showed a tendency to "click, click" when task saturated. One pilot summed up the concept: "The more complicated the 'button pushing' becomes, the sooner I disconnect the auto systems, including the autothrottles."

This reaction is certainly appropriate if the automation is about to lose control of the airplane or fly it into a mountain. But what if the severity of the situation isn't clear-cut and the automation failure isn't black and white? Then how can pilots detect the problem, devise a solution, and get the airplane back to where it needs to be? In some automation accidents, shedding the highest level of automation wasn't the right answer.

Case Study: Turkish TK-1951

On Feb. 25, 2009, Turkish Airlines Flight TK-1951 crashed short of Runway 18R at Amsterdam-Schiphol International Airport (EHAM), Netherlands. The Dutch Safety Board concluded that the accident "was the result of a convergence of circumstances" that included a malfunctioning radio altimeter. We might suspect a radio altimeter to play a pivotal role during a Category III ILS where it would guide the airplane to the runway during its last few feet in near zero-zero fog. But the weather, 1,000 to 2,500 ft. overcast ceilings with a visibility of 4,500 meters, was not a factor for Turkish TK-1951. So how could the radio altimeter have contributed to this crash?

The airplane was being flown from the right seat by a first officer receiving route training from a highly qualified captain. A second first officer sat in the jump seat as a safety observer. This Boeing 737-800 had two autopilots, two radio altimeters and a single autothrottle system. It was being flown using the right autopilot, as might be expected with the airplane being controlled from the right seat. The autopilots cannot be engaged simultaneously unless an ILS frequency is tuned and the approach mode is selected. The autothrottles are usable from takeoff to landing and automatically retard once the airplane is over the runway below 27 ft.

There are a variety of Boeing 737 autothrottle systems, most of which marry the autothrottle to the same-side radio altimeter as the controlling autopilot.

DUTCH GOVERNMENT

Turkish TK-1951, from Dutch Safety Board Report

But in the accident aircraft, the autothrottle was linked to the left radio altimeter unless it was not working. These connection variables were not documented in any manuals available to the crew.

The arriving flight was vectored in for a "short line up," bringing it onto final approach 3 mi. inside a normal intercept point. This would get the aircraft to the runway sooner but would require the pilots to join the glideslope from above.

During the approach, the left radio altimeter suddenly indicated an erroneous height of -8 ft. and may have caused the landing gear warning horn to activate prematurely. The crew acknowledged the radio altimeter as the possible cause, but did not discuss any further ramifications of the malfunction. These types of radio altimeter system faults were common in the fleet. In fact, Turkish Airlines documented 235 such faults in the previous year, including 16 on the accident aircraft.

Once the aircraft captured the localizer and glideslope, with the left radio altimeter reading -8 ft., the conditions to automatically reduce power for landing were met and the autothrottles brought both engines to idle thrust. The captain's flight display annunciated "RETARD" and his flight director steering bars disappeared. At that moment, the aircraft was nearly 2,000 ft. in the air.

Because the pilots were still descending steeply to join the glideslope and attempting to decelerate as they increased flap settings, the crew failed to notice that the throttles never increased from idle once they were on glidepath and on speed. They had several further clues but failed to notice the abnormally high deck angle as the right autopilot trimmed the stabilizer to maintain glideslope and the idle thrust allowed the airspeed to fall 34 kt. below target.

At 460 ft. above the ground the stick shaker activated. It took the startled crew almost 25 sec. to fully commit to the stall recovery, but by then it was too late. The airplane impacted 1.5 km short of the runway and broke into three pieces. Of the 135 souls on board, nine were killed.

The crew never considered level reversion because the highest level of automation, the right autopilot's capture of the localizer and glideslope, was working just fine. By the time the crew arrived at their "click, click" moment it was too late.

If you were to consider the automation as not just a collection of hardware and software but as another pilot, you would say this particular digital pilot became distracted by a minor systems glitch and failed to fly the airplane. With that set of blame fixed, you could also say the other three pilots on that flight deck failed in their duties to monitor. We need to start thinking of our seemingly infallible automation as just another pilot that can make mistakes, just like the rest of us.

New Approach to Digital Management

While our avionics have advanced significantly over the years, the same cannot be said of our automation philosophy. Perhaps it is time to treat the autopilot with the same respect and wariness we display toward any other pilot. Perhaps it is time to extend some of our best CRM skills to the automation. Call it Automation Resource Management (ARM).

We use CRM to continually support and monitor the other pilot. Why not ARM to continually support and monitor the flight automation systems? When not the pilot flying, a very good first officer mentally flies the airplane and is ready to offer the captain everything from gentle hints ("A little left of course, boss") to a slap in the face ("Pull up, now!"). But the CRM skill needed here isn't what you would find between captain and first officer. While the automation is usually very good, sometimes it can be a little "thick."

Let's call this somewhat dense digital pilot "Otto." If you think of yourself as a training captain and "Otto" as a newly hired first officer, you will have the right mindset. As a "noob," (a.k.a., "noobie" or new hire) he is very capable but doesn't always understand instructions as they are meant. Like any other noob, Otto can handle specifically assigned tasks, but even with these he needs to be watched. When pilots assume this role with the electrons, a five-step process is in order: decide, predict, execute, confirm and adjust.

Decide. For every action, determine ahead of time how the action should be accomplished. If, for example, you are assigning Otto the task of climbing from 5,000 to 6,000 ft., decide how that is best accomplished. For your aircraft this might be by entering the next altitude in the altitude select window, pressing the vertical speed button, and dialing a rate of 1,000 fpm.

Predict. Before actually turning Otto loose, anticipate the appropriate outcome. In our example, we expect the following steps: the next altitude to show up in the pilot's flight display, the vertical speed button to illuminate, the altitude hold indicator to extinguish, the vertical speed mode annunciated, the flight director to increase in pitch, the autothrottles to increase slightly, the airplane to climb at 1,000 fpm, the next altitude to capture about 100 ft. prior to level off, the vertical climb annunciator to extinguish, the aircraft to level off, the altitude hold annunciator to illuminate, and the autothrottles to reduce slightly.

Execute. When activating Otto, it can be helpful to involve more than just the tactile sensation of pressing the button or dialing in the desired setting. If we verbally announce the action ("Vertical speed, one-thousand") and point to the expected result (such as the pilot's flight display annunciator) we involve aural and visual senses to help confirm the action. This also helps the other pilot play his or her CRM/ARM role.

Confirm. Ensure the predicted outcome happens. This confirmation needs to encompass more than just the specific task but also any unintended consequences. For example, in our climb instruction, did the act of setting a new altitude and the vertical speed mode impact any previously entered vertical navigation command? Did our climb instruction take us off a "climb via" standard departure?

Adjust. If Otto fails to react as anticipated, give him a simpler way to accomplish the same objective. If that, too, fails, further simplify the task and be prepared to take control completely. If our 1,000-ft. altitude climb results in a correct pitch response but full climb thrust and rapidly increasing airspeed, it may be time to examine the command speed target. Perhaps the flight management system's speed profile was in error. You may need to overrule the autothrottles but still permit Otto to continue the level off.

Breaking down these mundane tasks into this five-step process may seem unnecessarily complicated, but it forces you to mentally fly the airplane before Otto does. Most of us already use four of these steps; we simply need to add the prediction element. But this process isn't enough for critical phases of flight. Here again we need to don our training captain caps when dealing Otto.

Automation Resource Management

DECIDE
What will the necessary inputs be?

PREDICT
What will the outcome be in terms of switch positions, lights, meters, annunciators and aircraft reaction?

EXECUTE
Activate the necessary controls.

Involve the rest of the crew by announcing the action.

CONFIRM
Point to the predicted reaction indicator (switch light, meter reading, indicator, pilot display annunciator or other indicator).

Verify the aircraft reacts as expected.

ADJUST
If the reaction is not anticipated, undo the last selected action.

Re-accomplish the decide/predict/execute/confirm/adjust process if possible.

Accomplish the desired action without automation.

MENTALLY FLY THE AIRPLANE DURING CRITICAL PHASES OF FLIGHT
Make control (pitch, roll, yaw, thrust) decisions in real time.

Evaluate performance (altitude, airspeed, vertical speed, heading, course).

If automation makes an unexpected control input or the resulting performance is wrong, be ready to overrule the electrons.

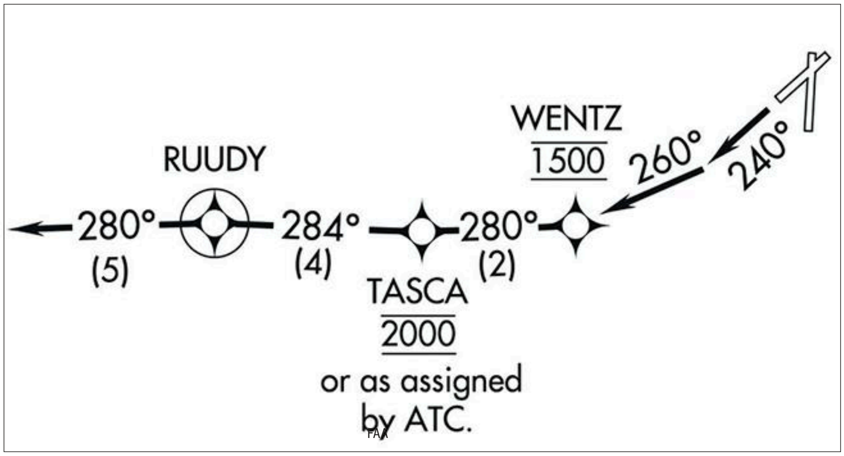
Mental Flight Required

During takeoff, initial climb, approach and landing Otto can easily fly the aircraft ahead of the pilot's brain. U.S. Air Force student pilots, new to the supersonic T-38 Talon, have a right of passage whereby their instructors sit them down after their initial sortie and say "Two, nine, nine, two." When the student asks what that means the instructor says, "You are so far behind the airplane, lieutenant, that your brain just crossed 18,000 ft. and we've been on the ground for an hour." As all those fledgling lieutenants come to know, 29.92 is the barometric setting when flying at or above FL 180. To the point, you cannot hope to fulfill your ARM role if you fall behind the aircraft during a critical phase of flight.

Let's say you are blasting off in your Gulfstream V from Teterboro Airport, New Jersey (KTEB) on the RUUDY FIVE departure. You are confident your lateral navigation (LNAV) system can maintain the 240-deg. heading to intercept the 260-deg. course to WENTZ, TASCA and RUUDY. You've never flown the procedure before, but it seems to be tailor-made for your aircraft's outstanding vertical navigation (VNAV) system to climb and cross WENTZ at 1,500 ft. and then TASCA at 2,000 ft. just 2.0 nm later. So you happily press the LNAV and VNAV buttons just prior to takeoff.

Once the gear and flaps are up, the aircraft pitches up to capture 200 KCAS and passing 1,000 ft. you have a climb rate in excess of 4,000 fpm. Sometimes the airplane does this, sometimes it doesn't. If you stayed mentally engaged during the takeoff would you have spotted this misbehavior as soon as the aircraft started to pitch up? As it turns out, this misbehavior catches GV pilots by surprise now and then and the 1,500-ft. mandatory altitude at WENTZ could make for a good case study in altitude busts.

Our climb example could result in a midair collision with traffic arriving at



Teterboro RUUDY FIVE departure, extracted from FAA SL-890, April 28, 2016.

nearby Newark Liberty International Airport (KEWR) but has thus far only resulted in a procedural violation or a stern reprimand from New York Departure Control. Unfortunately, aviation history is filled with case studies of approach and landing sequences in which pilot-automation disconnects had dire consequences.

For example, in 2004, two pilots failed to set the correct ILS frequency on approach to Houston-Hobby Airport, Texas (KHOU). The pair had flown a variety of aircraft, some of which had the glideslope pointer on the left of the attitude indicator; others had it on the right. On this approach, they mistook their "fast/slow" pointer to be the glideslope indicator and flew it centered until slamming into the ground 3 nm short of the runway.

While these pilots were not alone in seeing a glideslope pointer where it wasn't, most of these kinds of mishaps could have been prevented had the pilots mentally flown the raw data while monitoring the flight director's indications. This mental exercise is more than just, "Yeah, the needles are centered; I would have done that, too." The mental activity encompasses basic instrument flight procedures. For example, "I would use about 75% rpm and a 3-deg. pitch to maintain VREF when fully configured at this weight. Our nose is too high right now, what's going on?"

Giving Your Automation Epaulets

Today, many in the computer world eagerly anticipate the day artificial intelligence reigns. There is no doubt that many of our current generation airplanes have flight management systems that make real-time decisions with an accuracy rate better than can any pilot. But these systems can also make mistakes at a faster rate and it is up to the human being to catch and correct these before they become tragic. The "click, click" response can leave the pilot with not only the task of figuring out what's wrong and what to do about it but also having to do this while suddenly manipulating a handful of airplane.

The modern approach to this problem has been to ratchet down the automation level so the pilot can gradually assume increasing levels of control; this should minimize the "startle response." But automation failures rarely pop up as cut and dried failures. As the complexity of automated systems increases, error detection and troubleshooting become increasingly difficult. Merely removing the highest level of automation may not cure the system's ailment.

However, if we start to think of the automated systems as a fellow pilot, albeit a novice one, we can start to better manage the digital crewman. We know this novice is very good at executing a task when given very precise instructions but not so good at reading between the lines and is most certainly incapable of reading our minds. If we approach every automation task by thinking of ourselves as flight instructors and the electrons as students, we will be much better prepared for the times when the latter suddenly misbehaves. At that point, we will have wrapped our arms around ARM, Automation Resource Management. **BCA**